Igor Jakovljević, Bsc

Implementation of a Natural Language Processing Interface in the role of a Customer Support System and Agent

Master's Thesis

to achieve the university degree of

Master of Science

Master's degree programme: Computer Science

submitted to

Graz University of Technology

Supervisor

Assoc.Prof. Dr. Christian Gütl

Institute for Interactive Systems and Data Science (ISDS)

Head: Lindstaedt, Stefanie, Univ.-Prof. Dipl.-Inf. Dr.

Graz, October 2019

Igor Jakovljević, Bsc

Implementation einer maschinellen Sprachverarbeitungsschnittstelle als Kundenbetreuungssystem

Diplomarbeit
zur Erlangung des akademischen Grades
Master of Science
Masterstudium Computer Science

eingereicht an der

Graz University of Technology

Betreuer
Assoc.Prof. Dr. Christian Gütl

Institute for Interactive Systems and Data Science (ISDS)
Leitung: Lindstaedt, Stefanie, Univ.-Prof. Dipl.-Inf. Dr.

Graz, October 2019

Affidavit

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly indicated all material which has been quoted either literally or by content from the sources used. The text document uploaded to TUGRAZonline is identical to the present master's thesis.

Date

Signature

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe. Das in TUGRAZonline hochgeladene Textdokument ist mit der vorliegenden Dissertation identisch.

Datum

Unterschrift

Abstract

In the early nineteen-sixties Weizenbaum proposed a program concept that was able to converse with a human, this marked one of the first mentions of modern dialog systems. Since then dialog systems have found purpose in different areas of human life, ranging from chatbots that answer simple questions to complex dialog systems that retrieve information from other systems and present them to users in a meaningful and informative way. This led to the creation of the most fundamental question in the area of dialog systems, is it possible for a machine to exhibit and simulate human-like intelligent behavior while conversing with a human. In the early years of dialog systems, the use of pattern matching for recognition of user intent was primary, nowadays dialog systems are complex technologies that utilize the power of machine learning, natural language processing, and natural language understanding to interpret the intent of the user.

The goal of the thesis was to implement a prototype dialog system that is used as an information retrieval and customer support system for the company smaXtec. The prototype is used to determine if it is possible to create a dialog system that could integrate into the existing customer support communication channel and aid customers with their daily tasks. These tasks include finding answers to specific questions or retrieving relevant information from the systems knowledge base while reducing the workload of employees. Besides finding out if the Chat Agent System can replace the old smaXtec system, it was important to find out if the users would like to continue to use the new Chat Agent System. Determining how the users felt and which emotions did they experience while using the Chat Agent System was an important aspect of the thesis. The implemented prototype system can receive user input via a chat user interface, interpreter the user request, which is in English natural language and provides an answer to the user query. The interpretation of the user request is done twofold. At first, the user request is analyzed using natural language analysis and the result of the analysis is used to retrieve information. Then this information is formatted and presented to the user.

To evaluate the Chat Agent System it was necessary to design and conduct a user survey. The user survey was conducted in the headquarters of the company smaXtec with fifteen participants of whom ten were employees of the company and five were students from the Graz University of Technology. At the beginning of the study the

participants were introduced to Dialog Systems and the purpose of the new Chat Agent System, afterward they were given five tasks to execute with the use of the old smaXtec system and the new Chat Agent System. These tasks included tasks in the area of information search, information retrieval and tasks related to frequently asked questions. After the execution of the tasks, the system was evaluated using standard scales like the Computer Emotion Scale (CES) and the System Usability Scale (SUS). The results of the CES showed that the participants felt the emotion of Happiness during the usage of the Chat Agent System. On average the result of the System Usability Scale (SUS) produced a value of 78.66 points. While using the Chat Agent System the participants interacted on average 11.06 times with the chatbot to find information on preselected tasks. The user feedback on the prototype system had positive sentiment, with users stating that they would like to continue to use the Chat Agent System.

Kurzzusammenfassung

Anfang der sechziger Jahre schlug Weizenbaum ein Programmkonzept vor, das sich mit einem Menschen unterhalten konnte. Dies war eine der ersten Erwähnungen moderner Dialogsysteme. Seitdem haben Dialogsysteme in verschiedenen Bereichen des menschlichen Lebens einen Verwendungszweck gefunden, angefangen von Chatbots, die einfache Fragen beantworten, bis hin zu komplexen Dialogsystemen, die Informationen aus anderen Systemen abrufen und sie auf sinnvolle und informative Weise den Benutzern präsentieren. Dies führte zur Entstehung der grundlegendsten Frage im Bereich der Dialogsysteme: Ist es einer Maschine möglich, ein menschliches und intelligentes Verhalten während des Gesprächs mit einem Menschen zu zeigen und zu simulieren? In den Anfangsjahren von Dialogsystemen stand die Verwendung des Musterabgleichs zur Erkennung der Absicht des Benutzers im Vordergrund. Heutzutage sind Dialogsysteme komplexe Technologien, die die Kraft des maschinellen Lernens, der Verarbeitung natürlicher Sprache und des Verstehens natürlicher Sprache nutzen, um die Absicht des Benutzers zu deuten.

Ziel der Arbeit war es, ein Prototyp-Dialogsystem zu implementieren, das als Informationsabrufsystem und Kundenbetreuungssystem für das Unternehmen smaXtec verwendet wird. Der Prototyp wird verwendet, um zu bestimmen, ob es möglich ist, ein Dialogsystem zu erstellen, das in den vorhandenen Kommunikationskanal der Kundenbetreuung integriert werden und Kunden bei ihren täglichen Aufgaben unterstützen kann. Diese Aufgaben umfassen das Finden von Antworten auf spezifische Fragen oder das Abrufen relevanter Informationen aus der Wissensdatenbank des Systems, während die Arbeitsbelastung der Mitarbeiter verringert wird. Neben der Frage, ob das Chat Agent-System das alte smaXtec-System ersetzen kann, war es wichtig herauszufinden, ob die Benutzer das neue Chat Agent-System weiterhin verwenden möchten. Ein wichtiger Aspekt der Arbeit war es, herauszufinden, wie sich die Benutzer fühlten und welche Emotionen sie bei der Verwendung des Chat Agent-Systems verspürten. Das implementierte Prototypensystem kann Benutzereingaben über eine Chat-Benutzeroberfläche empfangen, die Benutzeranfrage interpretieren, die in englischer natürlicher Sprache ist, und eine Antwort auf die Benutzeranfrage bereitstellen. Die Interpretation der Benutzeranfrage erfolgt zweifach. Zuerst wird die Benutzeranfrage mithilfe einer Analyse der natürlichen Sprache analysiert und das Ergebnis der Analyse wird zum Informationsabruf verwendet. Dann werden diese Informationen formatiert und dem Benutzer präsentiert.

Um das Chat Agent-System zu bewerten, war es notwendig, eine Benutzerbefragung zu entwerfen und durchzuführen. Die Benutzerbefragung wurde im Hauptsitz des Unternehmens smaXtec mit fünfzehn Teilnehmern durchgeführt, von denen zehn Mitarbeiter des Unternehmens und fünf Studenten der Technischen Universität Graz waren. Zu Beginn der Studie wurden die Teilnehmer mit Dialogsystemen und dem Zweck des neuen Chat Agent-Systems bekannt gemacht. Danach erhielten sie fünf Aufgaben, die sie unter Verwendung des alten smaXtec-Systems und des neuen Chat Agent-Systems ausführen sollten. Diese Aufgaben umfassten Aufgaben im Bereich der Informationssuche, des Informationsabrufs und Aufgaben im Zusammenhang mit häufig gestellten Fragen. Nach der Ausführung der Aufgaben wurde das System anhand von Standardskalen wie der Skala der Emotionen im Zusammenhang mit dem Erlernen neuer Software (Computer Emotion Scale - CES) und der Skala der Systembenutzbarkeit (System Usability Scale -SUS) bewertet. Die Ergebnisse der CES zeigten, dass die Teilnehmer während der Nutzung des Chat Agent-Systems die Emotion des Glücks verspürten. Im Durchschnitt ergab das Ergebnis der Skala der Systembenutzbarkeit (SUS) einen Wert von 78,66 Punkten. Während der Nutzung des Chat Agent-Systems interagierten die Teilnehmer durchschnittlich 11,06 Mal mit dem Chatbot, um Informationen zu vorausgewählten Aufgaben zu finden. Das Benutzer-Feedback zum Prototypensystem war positiv und die Benutzer sagten, dass sie das Chat Agent-System weiterhin benutzen möchten.

Contents

List of Figures

List of Tables

1. Introduction

In the early nineteen-eighties the first customer relationship management (CRM) system became visible. One of the early systems was created by Michael Wilke and Robert Thornton. The system was a Web browser based contact management system that offered e-consumer affairs capabilities including consumer demographics, intelligent workflow, issue escalation, base knowledge, survey scripting, order entry, campaign management, and consumer insight reporting. Samsudin & Juhary (2014) have shown that many authors agree that a CRM system cannot be strictly defined, but that there are two main approaches to the definition of CRM system. One definition is from the perspective of management. Meaning that it is an approach to identify, acquire and retain customers (Ellatif, 2008). The second perspective is from the perspective of information technology, a CRM system can be defined as a tool or a system to support relationships strategies and activities such as identifying, acquiring, and retaining customers (Chen & Popovich, 2003).

A customer relationship system is a specialized CRM software that is used to handle interactions between the company and the customer. The integration of modern technologies like telephone or internet web pages has been adopted in modern customer relationship systems. Through these integrated technologies, customers can directly communicate with the CRM software. This has popularized this modern CRM software because they provide a way for the customers to contribute to the improvement of the quality of the product which increases the profits of the company. These systems greatly improved the work and processes in companies by providing structure and digitization to already existing customer relationships systems (Goodman & Wilke, 2007). Xu et al. (2017) have shown that social media has changed the way that the modern CRM system works. Customers are using social media in order to get help, as they can easily send a chat message or Facebook status rather than draft a detailed email (Nielsen, 2011). The increased interest of customers to use social media for the communication with the company, in particular chat based systems, has produced a special communication channel. This communication channel is called a chat support communication channel. It combines functionalities from two different communication channel, the speed of telephone chat and the readability and data persistence of e-mail based communication channel. Customers send millions of chat messages to major brands on a monthly basis in the U.S. alone. With the amount of user requests increasing, it has become difficult to process and respond to them. The huge number of customer requests forced organizations to deal with this problem. Many organizations have created teams for customer service intending to respond to a large number of requests. These teams are called customer service teams, and the size of these teams range from a couple of employees to a couple of hundred of employees trained and educated to meet the need of the users. (Murray, 1991). However, educating personnel and answering these requests manually is a time-consuming and costly task. Since the early 1960s chatbots and dialog systems have been used to simulate human interaction and execute information retrieval tasks (van Woudenberg, 2014). With the rise of artificial intelligence and deep learning technologies, the customer relationship system community was impacted with the desire to integrate smart algorithms into existing software to better understand the customer, reduce the cost of personnel training and reduce the response time of support agents. Chatbots are a replacement for real personnel in the area of tracing and recovery of specific information. They also found their usage in data retrieval by analyzing users request, that have been generated in natural language, and converting them into a query or queries to retrieve data with well defined structure and semantics.

Chatbots have not only found a role in the traditional CRM systems, but have also started being used as extensions or replacements to Web pages and systems for information retrieval. Asadi & Hemadi (2018) show how chatbots can be used in e-commerce Web pages to help the user. The user has the ability to retrieve information about his/her basket, items that he/she wants to buy, account information via a chatbot. This same chatbot is also used to answer general questions about the company (e.g. shipping time, return policy, etc.). In early 2018, a virtual assistant with the name of Leo was developed in the Department of Public Affairs of Riga Technical University (RTU). Better communication and reduction of effort for providing information to the students, new or existing, was the main goal of the project. This chatbot was able to retrieve profile information about students and reduce the time needed to get personal information from the system (Mislevics et al., 2018). The students did not have to navigate to the web page of the university and search for the information. Instead they ask the chatbot to retrieve the information. The YPA is an example where a chatbot is used as an information retrieval system. The retrieval of information from the British Telecom's Yellow pages is the functionality of the YPA dialog system implemented in natural language. The yellow pages contain advertisements, with the advertiser name, and contact information. YPA is used to retrieve addresses of the advertisers. In the case that no address is found, a conversation is started. The system asks users more details in order to give the user the required address (Kruschwitz et al., 1999). Lasek & Jessa (2013) introduced a program which simulates an intelligent conversation with web page visitors, dedicated to hotels and guesthouses. This chatbot was intended as a guide through the web page, showing users where to click in order to get information or to book rooms. Technology giants like Microsoft, Apple and Amazon have also noticed the trend of chatbots and have implemented their own platforms. These companies have focused on creating personal assistants performing various tasks in order to help the user throughout the day (López et al., 2018). These chatbot have the ability to retrieve the schedule of the user, weather information, read emails and much more. One of the main characteristic of chatbots powered by technological giants is that they are voice based personal assistants and that they try to simulate conversation in real time with the user. These examples show how chatbots can be used not only as extensions to CRM systems, but also as means to improve the user journey through an information retrieval system or a company web page. This can be achieved by using the concept of personal assistants on web pages in order to facilitate information retrieval.

One of the challenging problems in the field of artificial intelligence (AI) is the creation of a chatbot engine that communicates with users in natural language. Bergner & Johannsen (2016) describe how natural language processing (NLP) can be used to improve business processes and interaction with customers by analyzing their statements to provide feedback. Chatbots can also be used to convert user questions to database queries and retrieve data directly without the usage of an additional application. DocChat is a chatbot that offers the users data retrieval and question answering capabilities. Unlike traditional chatbots that provide question answering capabilities based on pattern matching, DocChat has the capability to convert the question of the user into a structured query for information and data retrieval (Yan et al., 2016).

The social media trend and the fast development of chatbot technologies has created the opportunity for the utilization of chatbots as replacements for information and data retrieval applications. This development has led to the switch in paradigm, where companies want to replace existing applications that are used for information retrieval with chatbot applications. These chatbot applications provide prompt and personal response to the customer, replacing or extending traditional user interfaces. Microsoft CEO Satya Nadella said, "Chatbots are the new apps" (della Cava, 2016). Chatbots are seen as a medium for direct customer engagement through text messaging for customer service, bypassing the necessity for special-purpose applications or Web pages. This transition from traditional Web services and Web pages to chatbot based apps is not simple. Despite the difficulty of the transition, Patrick Zimmerman predicted that by the year 2020, more than 75 % of all companies will have a chatbot application present as a way to interact with customers (Zimmermann, 2019).

Cui et al. (2017) present an idea of a chatbot based customer support agent that is integrated into an existing business with the goal to improve the existing communication channel and help the user retrieve information. The information that the user is retrieving is information that was previously retrieved by a information retrieval system or a web application. This use case revealed that chatbot applications can be used as replacement for multiple aspects of CRM systems and also serve as a integral system component of a information retrieval system of a company. The ever-growing and evolving CRM systems and the emerging chatbot technologies make it possible to combine these two concepts into a new feature that is cost effective and provides an improvement to the customer experience. As seen in this chapter chatbots can be used to replace existing web page, but also to improve or replace communication channels of CRM systems (Abu Shawar & Atwell, 2007).

1.1. Motivation and Background

The company smaXtec[1]is based in Graz and is specializing in comprehensive solutions for monitoring cow herds. It was founded in 2009 with a clear commitment to bring long-term positive change to the dairy industry. In just a few years of the company's existence, it has grown to be a large international competitor in the field of dairy production, having sales representatives in countries like Brasil, Ireland, Germany and many more. The main product that the company is based around is the smaXtec bolus. The smaXtec bolus[2] measures the activity, temperature, and ph values in cows. Besides the basic smaXtec bolus, a premium version is also available that gathers additional pH data.

The growing trend of conversational agents made the company representatives recognize that improvement to their communication channels was necessary. With the growth of the company, the task of answering simple questions that are repeated often (e.g. "How does the smaXtec activity measurement work?") is becoming ever more costly. The usage of the frequently asked questions page does not provide the customer with the satisfaction that he/she would experience when talking to a sales representative or a human support agent. Keeping that in mind, the company representatives searched for a cost-effective solution in order to increase the satisfaction of the customers with an emphasis on preserving the quality of the answers. Their main desire is to improve the current customer support system with a new chatbot and dialog system technologies. The existing communication channel of the company consists of an email system, intercom system[3] and web-based solution that hosts frequently asked questions. It is extremely hard to maintain a system that is divided between three different platforms. Each system has its positive sides, but the common

[1] http://www.smaxtec.com
[2] https://www.smaxtec.com/en/smaxtec-classic-bolus-activation/
[3] https://www.intercom.com

failure point is the answering speed and the knowledge domain of the support agent. Support agents with a large knowledge domain are not cost effective for a company in growth. Support agents with a high response rate and low knowledge domain lower the customer satisfaction level. The company also uses an additional service called smaXtec messenger[4] where the customers have the ability to retrieve information about the status of their herd and individual animals of the herd.

Since the nineteen-eighties CRM systems have hugely evolved into sophisticated software solutions offering the company that is using them a huge number of features, ranging from transcripts of customer conversations with sales representatives to product suggestions for new customers based on previously gathered data (Samsudin & Juhary, 2014). Chakrabarti & Luge (2015) inform that many businesses have seized the growth of popularity of chat agents as a communication channel and already implemented customer support areas through these chat agents. Recently the use of chat agents as customer service representatives has skyrocketed in popularity. The chat agents do not need rest and do not get tired as human customer service representatives. Besides that, chat agents have a large information database and have the ability to quickly access that information, which is lucrative and appealing to companies.

The American company Facebook used the opportunity of the rise in popularity of chatbots to expand its messenger platform to offer chat agents and chatbots to diverse businesses. Later Amazon, Microsoft, and many others joined Facebook and created their own platforms for chat agents and chatbots. Customer support that is powered by humans is a key tool for growing companies to improve their product and keep the customer satisfaction level at a high point. With the growth of the company, the costs to keep the system running increase greatly, thus decreasing the ability of the company to scale up the support system with the growth of the company. According to IBM's research, chatbots and chat agents provide a great number of benefits to growing companies Schneider (2017). These benefits can be summed up into pre-emptive action, reduction in training time, customer service that is active 24/7 and scalability. No human can achieve the level of response as an automated system. Even though a chatbot cannot currently resolve all customer queries, it can be used to resolve many frequently repeated queries that typically make up most of the user requests. Based on Schneider (2017) the ability of chatbots to resolve simple customer queries that occur often, shows a huge potential for companies. It could lower the abandonment of customers significantly, while keeping customer satisfaction on a high level. Looking at the fact that employing and educating staff requires time and monetary investment, these costs can instantly increase with an increase in staff numbers. Based on a survey CareerBuilder (2017) of over 2300 managers and HR professions, nearly 75 percent

[4]https://www.smaxtec.com/en/smaxtec-messenger/

agree that the on-boarding process lasts approximately one month. Cost and time reduction is huge benefit of automation platforms. Schneider (2017) states that the IBM Watson platform comes already predefined with industry and also domain knowledge and that it required only one training process. According to Risueño (n.d.) a chatbot can be trained within 2 days to provide answers to as much as one million different base sentences. Based on Eurostat (2016) survey carried out in Europe, a full-time employee in the EU works 40.3 hours per week. Compared to that, automated customer service offers a 24 hour workday and does not have the constraint of time zones or public holidays. This in terms means that the customers can contact chat based customer support at any time and can have their inquiries resolved. As shown in the previous points, the ability to train a chatbot faster than an employee (Eurostat, 2016) and (CareerBuilder, 2017), enables more scalability to chatbot based systems. When starting to use a chatbot, business immediately have FAQs systems, that can help resolve thousands or more queries from the customer without the use of a customer support employee. These businesses can respond to a large increase of customer queries simply by switching to a more powerfull server machine. Artificial intelligence enables that chatbots to constantly learn from every interaction, this consequently means that businesses can expand to new markets without the boundaries of the speech barrier and employment of local staff which requires additional training. Even though a chatbot cannot correctly interpret all customer queries, it can be used to resolve many frequently repeated queries that typically make up most of the user requests.

The benefits that chatbots provide in the field of customer support are also visible in the field of information retrieval. Customers lean more to use messaging platforms in order to solve issues than web-based solutions. In figure 1.1 it is visible that 84.6% of all customers prefer a chatbot communication channel to get answers to simple questions. Based on the fact that customers appreciate chat based channels more than web-based channels, the idea forms that chatbots have the potential to replace Web pages and web services (Valtolina et al., 2018).

Figure 1.2 shows the results of an experiment, where the users were given the task to retrieve information from a Web page and from a chatbot application. When using the chatbot application, all users were able to complete the information retrieval task. With the use of the Web page 57% of the users were not able to complete their task. The main reason for their failure was that the information was not properly displayed on the Web page due to the position of different graphical elements. These restrictions did not apply to the chatbot application (Valtolina et al., 2018).

Taking into consideration the results of the above mentioned experiment, the benefits of chatbots in customer support systems and the desired communication channels of customers it can be seen that chatbot systems could be used as a replacement for already existing platforms. In the case of the company smaXtec, features of the smaXtec

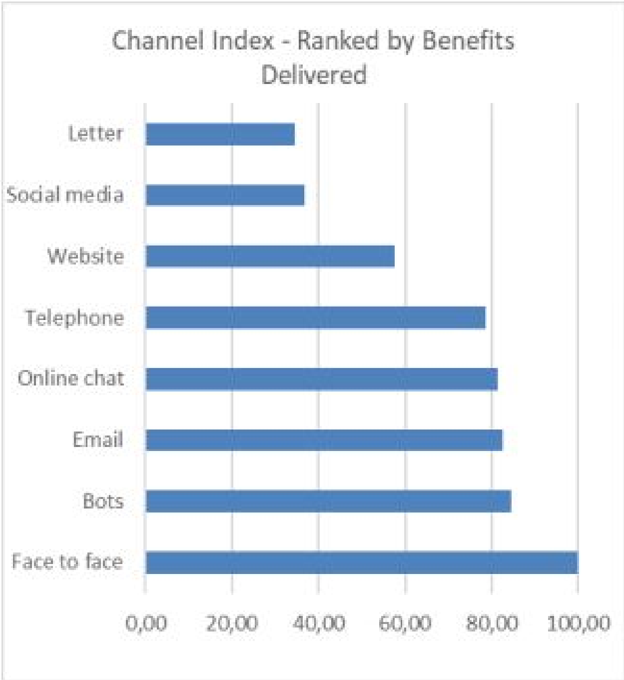

Figure 1.1.: Users' Preferred Communication Channel (My Clever Agency, 2018)

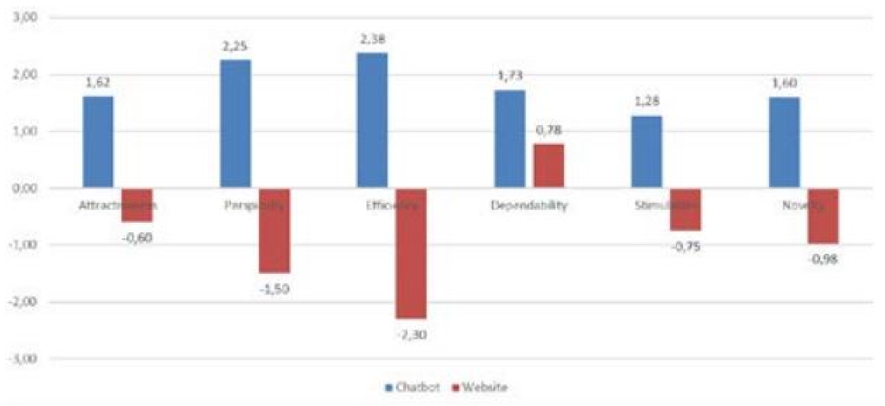

Figure 1.2.: Chatbot And Website Usage Compared (Valtolina et al., 2018)

messenger platform can be reproduced with an chatbot system. This could reduce the time the customer has to seek for information about the company and about his/her own data (Valtolina et al., 2018; Brandtzaeg & Følstad, 2017).

Referring to the above mentioned situation, the main goal of this master thesis is to create a cost-effective system that is able to handle the increased number of customer queries. The system should leverage the growing trend of chatbot based communication channels in customer support systems. The future vision of the company is an advanced chatbot system that improves their communication channels and speeds up customer responses. Taking into consideration time and resource restrictions of the thesis, the end-product of the thesis will be a chatbot system that provides basic functionalities of the envisioned system. The prototype will be able to retrieve user information and answer simple queries. These queries will be based on the frequently asked questions, that were created from user inquiries. User information will be limited to basic information that is stored in the internal databases of the company.

1.2. Outline

The thesis is constructed in a way that it first focuses on the motivation and theoretical background that was necessary for the readers to understand the concepts that were discussed and used in the second part of the thesis. The second part of the thesis is the implementation and evaluation part, where the methods and technologies used for the development of the Chat Agent Systems are discussed. This discussion is followed by the user study setup together with the evaluation of the user study.

The thesis starts with an overview of the existing smaXtec system and the introduction to the motivation for the implementation of a new chatbot based system. This new chatbot based system has the goal to improve the existing smaXtec system. Clear goals for the thesis and the goals of the company representatives are defined in chapter 2.

Chapters 3 focuses on the historical development of chatbot and dialog systems together with emerging technologies that were used in these systems. Based on this chapter, chapter 4 introduces different categorizations and different applications for chatbots and dialog systems.

Functional requirements and non-functional requirements with a detailed overview of the smaXtec system are described at the start of chapter 5. Based on the previous chapters the selection of technologies and the formulation of the prototype system is explained in this chapter also. The last aspect of chapter 5 is the formulation of the conceptual architecture of the prototype and the description of integration between the smaXtec system and the prototype system.

Based on Chapter 5, the implementation of the suggested prototype is documented in chapter 6. The technologies used for the implementation of the prototype are described in detail, together with the main use-cases of the prototype system.

Chapter 7 describes the user's study that was applied to test, if the prototype system is accepted by future users. This chapter contains detailed descriptions of used questionnaires and scales like the SUS and CES scale. The goal of the user study is defined in this chapter. Another focus of this chapters is the discussion of the results and the evaluation of the outcome of the user study.

Lessons learned in the area of literature survey, implementation and overall process of the project are discussed in chapter 8. The outcome of the project together with ideas for future developments of the prototype system is discussed in chapter 9.

2. Main Objectives and Overview of the Existing System

The focus of the thesis is described in this chapter by stating the main objectives and providing a detailed overview of the technologies used in the current company system. A approach on how to integrate a chatbot communication channel in both the customer support channels and traditional web applications as a information retrieval unit has been suggested.

2.1. Main Objectives and Overview of the Existing System

To formulate the main objectives of the thesis, it is necessary to take into account the objectives and goals that were stated by the company representatives. The goal of the company is to create a system that integrates into their communication channels and is able to analyze, evaluate and provide an answer to questions like „Hey, what do I have to do on my farm today?", "There is a sticker with a date on each bolus. what does it mean?" and more. Besides the informational frequently asked questions, the chatbot has to be able to retrieve information from the existing back-end system about the animals of the customer and provide information summaries about them in a natural language form. The resulting system should speed up the response time of the customer support agent while keeping the answer quality on a high level. The main objectives stated by the company correlate with the described usage of chatbots in chapter 1.

To have a better understanding of the company communication channels, all systems have been analyzed to discover the benefits and points of failures of the systems. The communication channels of the company consist of the following four components an email system, an intercom system, a static web system, and a web application. Each communication channel has been analyzed and the summary, benefits and concerns are shown in table 2.1. Besides the communication channels of the company, also the internal software architecture was analyzed to discover the best way to integrate a chatbot system into the existing system.

System	Summary	Benefits	Concerns
Email System	The email system is the network of computers handling electronic mail (email) on the Internet.	Essential files can be attached. Cost-effective Medium Customer satisfaction surveys can be conducted.	Time Consuming Security Issues Misunderstanding
Intercom System	Intercom provides a chat web user interface that enables direct communication between the customers and employees.	All the communication with users and customers in one place	Hard to understand the differentiation between their product lines
Web System	Frequently asked questions (FAQ) are questions and answers that are asked daily and are related to a specific topic.	Content is easily manipulated and updated, Low Price	The content is static and customers cannot interact with a real person
Web Application	Web interface that the customers use for information retrieval.	Accessible from any personal computer or mobile device	It has a steep learning curve and it hard to understand for new customers

Table 2.1.: Benefits and Concerns of the Existing Communication Channels

Table 2.1 describes all communication channels of the company, but the thesis will focus on two main components the web system and the web application, with the emphasis on improving them with a system that simulates the ease of communication of the intercom and email system.

What makes smaXtec competitive on the market is that they introduced a separation of concerns in their software architecture as seen in figure 2.1. This separation of concerns enables easy integration of new components and maintainability of existing ones. The software architecture was split up in three main layers. The first layer and most visible one is the front end layer of the architecture. This layer covers the implementation of graphical interfaces that are shown to the customer. The programming language of this area is Javascript[1]. The products of this layer are the mobile applications and the web application. The layer of the software architecture that enables access to the data and interaction with the database is the back end layer. This layer has a focus on the Python

[1]https://www.javascript.com/

programming language[2]. Lastly, the layer that interprets data and produces value for the customer is the data analysis layer. The main goal of this layer is to examine the persisted data and find patterns in it. These patterns should be beneficial to the customer. This layer has also a strong focus on the Python programming language. Based on the inputs of the company representatives and the analysis of the company's

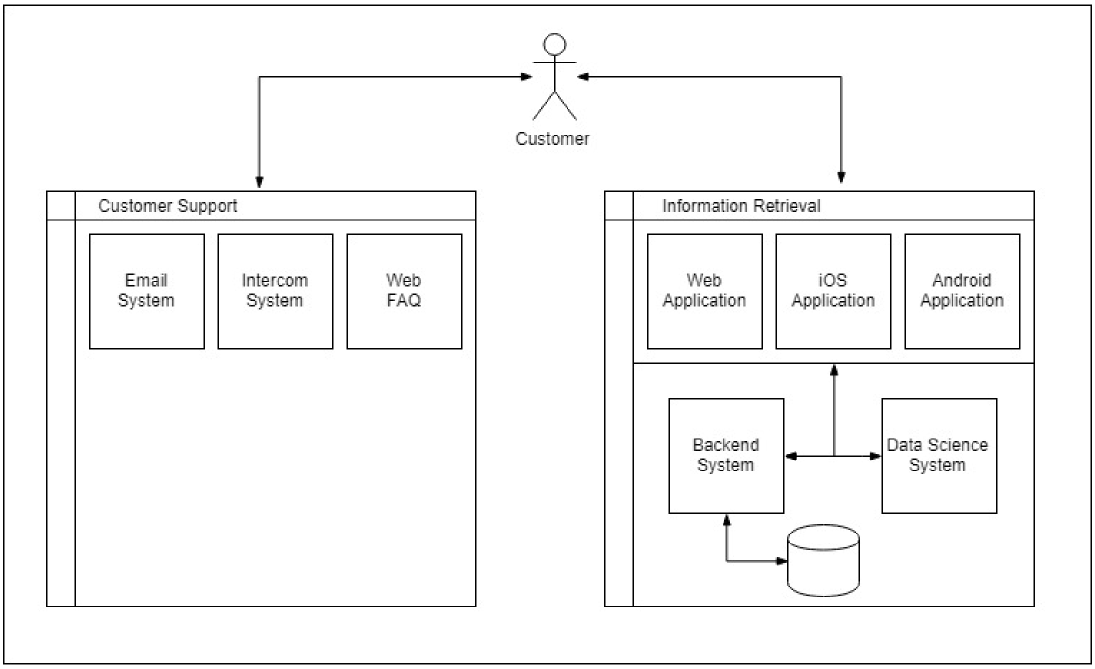

Figure 2.1.: smaXtec System

system the goal and objective of this thesis is a working chatbot system demonstrator that uses existing technologies and/or improves the communication channels of the company. To cover most of the companies customers, the resulting chatbot system should be available only in English. Due to the time restrictions of the thesis, the result will be a prototype system with the focus on retrieving user information and answering frequently asked questions. This system will have the ability to be integrated into the existing system of the company and also work as a standalone system. This implies that the same technological base that the company uses is used for the prototype. The integration of the chatbot system will not be part of the thesis, only the creation of a system that can be integrated will be covered by the thesis. Referring to the above mentioned, the main goal of the thesis is to create a prototype that demonstrates and confirms that the goals set by the company representatives can be achieved with the use of a chatbot system in their communication channels.

[2]https://www.python.org/

2.2. Extending the Current System

In order to extend the current system and integrate a chatbot communication channel, it is necessary to create an additional component. The component should be capable to communicate with the existing information retrieval system. This enables the system to retrieve information and data from the database. The chatbot component has to be displayed to the user through one of the front end system components or as a stand-alone front end component. This component will provide answers to frequently asked questions by the users. Which will act as a addition to the used customer support channels. It will also enable users to retrieve information about their own farms and cows. The frequently asked questions are formulated and asked through one of the three existing communication channels shown in table 2.1. The frequently asked questions, will contain questions from emails, questions formulated by company staff and intercom system chat log. The extension of the current system can be seen in figure 2.2.

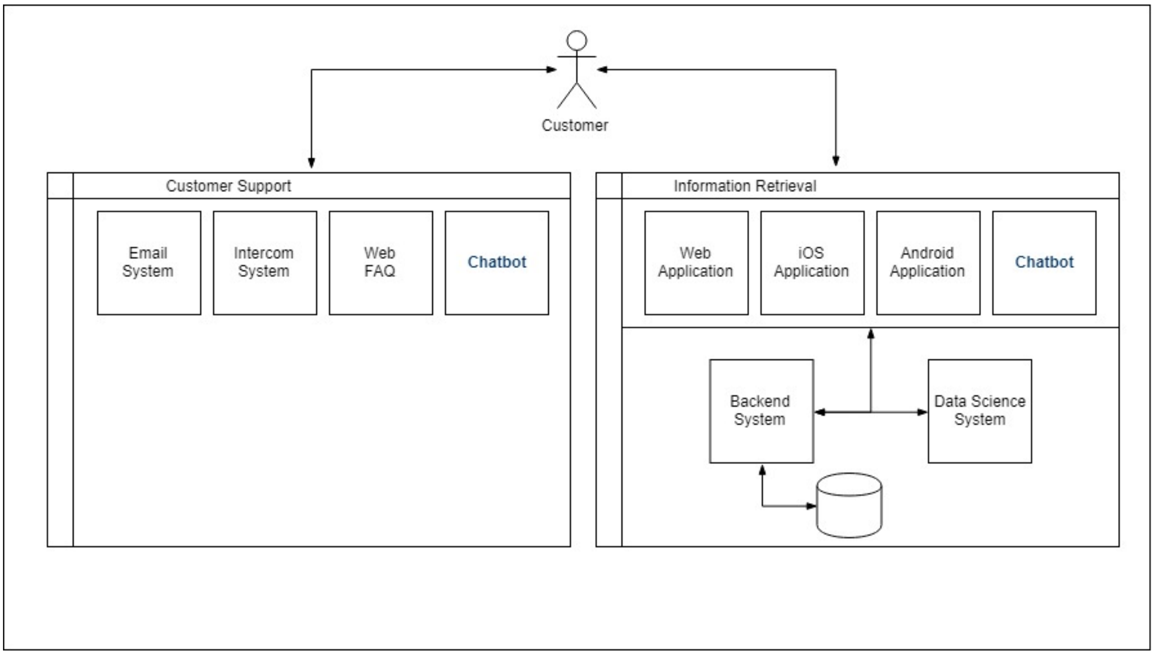

Figure 2.2.: smaXtec System - Extended With a Chatbot System

The chatbot system can be seen both as part of the customer support system and the information retrieval system. Figure 2.2 describes how the chatbot system has to be integrated into the current system. In order for the chatbot to be a stand-alone application, it is necessary that the system can communicate with the data science system and the backend system. Beside these information sources, the system also has a stand-alone database as a part of the system. The purpose of the database is to store the question-answer pairs. The integration of the chatbot system into the customer support system is enabled by the database. The unique architecture of the chatbot system is the foundation, that enables it to be used as a communication channel and an information retrieval system.

2.3. Summary

In this chapter, the goal of the thesis was established and an structure of the communication channels of the existing CRM system and architecture was described in details. As shown above, the current support system is based on employees using existing communication software packages to communicate with the customers. The goal of these communication packages is to answer customer queries as fast as possible with a high degree of reliability.

Based on the knowledge from chapter 1, it is visible that the current system can be improved with the use of a chatbot support agent. The chatbot prototype system should leverage used technologies and the existing system in order to validate that the goals of the company representatives stated in chapter 1 can be implemented. The prototype system should be easily integrated into the existing system of the company.

3. Conversational Agents and Dialog Systems

The focus of this chapter is the introduction to the topic of conversational agents and dialog systems. Besides these topics, natural language processing and the history of artificial intelligence and the usage of it in the area of natural language processing are described in detail. This chapter also includes an introduction to machine learning and a definition of popular approaches in the area of machine learning.

3.1. History

The origin of chatbot systems can be tracked back to 1960s ELIZA Program, that was developed by Joseph Weizenbaum (van Woudenberg, 2014). ELIZA simulates a psychologist and is capable of conversations as shown in the following listing 3.1.

```
User:Men are all alike.
ELIZA:IN WHAT WAY
User: They are always bugging us about something or other.
ELIZA: CAN YOU THINK OF A SPECIFIC EXAMPLE
User: Well, my boyfriend made me come here.
ELIZA: YOUR BOYFRIEND MADE YOU COME HERE
User: He says I am depressed much of the time.
ELIZA: I AM SORRY TO HEAR YOU ARE DEPRESSED
```

Listing 3.1: ELIZA A Computer Program For the Study of Natural Language Communication Between Man And Machine (van Woudenberg, 2014)

Looking at this conversation, ELIZA can be understood as an intelligent person, but ELIZA works on a very simple principle of pattern matching and substitution.

The creation of Joseph Weizenbaum's ELIZA was greatly influenced by Alan Turing's famous test. In the 1950s Alan Turing proposed a test as a criterion of intelligence in the influential essay "Computing Machinery and Intelligence". The Turing test is a test that assesses the ability of a machine to exhibit and simulate human-like intelligent behavior. Turing proposed that a human evaluator judges text-based conversation in natural language between a human and a machine that is created to generate human-like responses. The evaluator would be aware that one of the subjects engaging in the conversation is a human and the other is a machine, then based on the conversation he would have to judge which of the subjects is a machine and which is a human (Turing, 1950).

Complementing ELIZA which adopted the character of a Rogerian Therapist, PARRY a Chatbot mimics the behavior of a paranoid schizophrenic. PARRY works by using pattern matching and canned responses Colby & Gilbert (1966). JABBERWACKY was the next big breakthrough, it was the Chatbot created in 1988 by Rollo Carpenter. With the rise of technology and speech synthesis, Dr.Sbaitso was created and released in the late 1992 by Creative Labs. Dr. Sbaitso is an artificial intelligence speech synthesis program. By the end of the 20th century A.L.I.C.E (Artificial Linguistic Internet Computer Entity) was created by Richard Wallace (S. A. G. D. D. M. Deshpande Aditya & Joshi, 2017). A.L.I.C.E is a natural language processing bot that interacts with a human by applying pattern matching rules to provide an answer to a user query in a conversation.

3.1.1. Terminology

The terminology "Chatterbox" and "Chatbot" can be traced to a game character *Rog-O-Matic* for a multiuser dungeon game called *Rogue: Exploring the Dungeons of Doom*, developed by Michael Toy and Glenn Wichman. The task of the in-game character was to answer user inquiries concerning the navigation through the dungeon, other gamers and objects available in the game world (Kluwer, 2011) .

3.1.2. Turing Test and the Imitation Game

Turing (1950) questioned the ability of machines to think. The main question that was mentioned in that paper was "Can machines think?". Which caused the formulation of many problems that would reference that question. The main problem that Alan Turing formulated was the "Imitation Game". The "Imitation Game" is a problem that can be formulated in means of a game. It is played by three persons a man (person A), a woman (person B) and an interrogator (Person C) who may be of either sex as seen in figure 3.1. The task of the interrogator is to determine which person is a woman and which person is a man while being in separate rooms. Not being able to see the other persons and not knowing who is he talking to makes it hard to determine the identity of the individuals. The interrogator is able to write questions to person A or B and the persons have to answer the questions. The task of Persons A and B is to try to confuse the interrogator and answer the questions to convince him that they are their opposite. At the end of the game, the interrogator should try to determine who is a man or a woman (Turing, 1950).

From the formulation of the Imitation Game, Alan Turing questioned what would happen if a machine would participate in the imitation game instead of Person A or Person B. A simple diagram of the imitation game can be seen in figure 3.1. The question "What will happen when a machine takes the part of A in this game?" was formulated from this game(Turing, 1950). The introduction of this question caused the creation of the famous Turing Test, which questions the ability of the machine to answer questions as a human. The Turing test is now Widely used as a criterion to determine the "intelligence" of machines (e.g. Leobner Price for chatbots). The simplicity of the Turing test gives the test not only tractability but also breadth of subject matter while attaining emphasis on emotional and aesthetic intelligence. The Turing test cannot directly compare if the machine behaves intelligently, it can only test if the computer behaves like a human being.

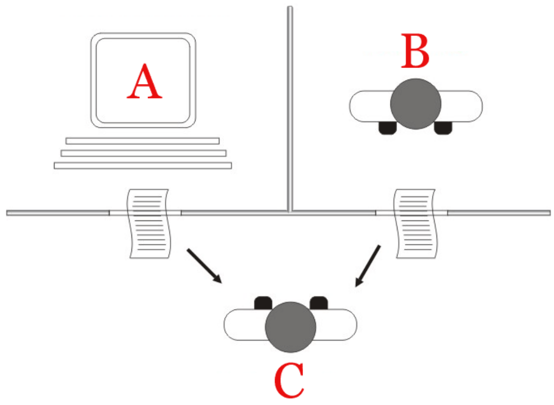

Figure 3.1.: The Turing Test Diagram (Margallo, 2007)

3.1.3. Loebner Prize

A relevant competition in the field of artificial intelligence and chatbots is the Loebner Prize. Which awards prizes to chatbots who are considered the most human-like. ELIZA and A.L.I.C.E. are some of the winners of this competition. It is an annual competition with the goal to find the most human-like chatbot. A silver medal and 25,000 USD is awarded to the first system that can convince most judges to be human. A winner of the gold medal and 100,000 USD should be additionally able to process audio and video input (Bradeško & Mladenić, 2012). All the winners of the competition are listed in figure 3.2. What needs to be mentioned is that the winner is always awarded relative to the other competitors.

Year	CA Name	Developer
1991	PC Therapist	Joseph Weintraub
1992	PC Therapist	Joseph Weintraub
1993	PC Therapist	Joseph Weintraub
1994	TIPS	Thom Whalen
1995	PC Therapist	Joseph Weintraub
1996	HeX	Jason Hutchens
1997	Converse	David Levy
1998	Albert One	Robby Garner
1999	Albert One	Robby Garner
2000	A.L.I.C.E	Richard Wallace
2001	A.L.I.C.E	Richard Wallace
2002	Ella	Kevin Copple
2003	Jabberwock	Juergen Pirner
2004	A.L.I.C.E	Richard Wallace
2005	George (Jabberwacky)	Rollo Carpenter
2006	Joan (Jabberwacky)	Rollo Carpenter
2007	UltraHAL	Robert Medeksza
2008	Elbot	Fred Roberts
2009	Do-Much-More	David Levy
2010	Suzette	Bruce Wilcox
2011	Rosette	Bruce Wilcox
2012	Chip Vivant	Mohan Embar
2013	Mitsuku	Steve Worswick
2014	Rose	Bruce Wilcox
2015	Rose	Bruce Wilcox
2016	Mitsuku	Steve Worswick

Figure 3.2.: Winners of the Loebner Prize since 1991, adopted from (Abdul-Kader & Woods, 2015)

3.2. Chatbot Systems

Sharma & Malik (2017) defines a chatbot as a computer application which handles a conversation through acoustic or textual methods. Terms like chatterbot, Conversational Agent, Conversational Entity are used as equivalents to the term chatbot. To complete the Turing test with a high score, chatbot programs are built with the intent to imitate the behavior of humans when executing such a test. The use of chatbots can be usually seen in dialog systems, and the functionalities of chatbots in these systems are vast. (C.P. Shabariram & Vidhya, 2017). These include customer service, information acquisition, or information and retrieval. Chatbots browse for keywords within the user input, before producing a reply for the most matching keywords, or the best pattern. Other use advanced natural language processing methods to produce acceptable output. This reply is usually loaded from a database, but in some cases, natural language understanding is used to generate responses (Kluwer, 2011).

3.2.1. Introduction to Chatbot Systems

Taking into consideration the history of chatbots, we can derive two main types of chatbot systems and architectures. The first one is the already mentioned ELIZA and the second one is the A.L.I.C.E. system (B. Shawar & Atwell, 2002).

3.2.2. ELIZA System

The way that ELIZA works was briefly mentioned in chapter 1, tthe system relies on simple pattern matching, based on a motive and answer model to process input and transform it into a suitable output. According to Weizenbaum (1966), the ELIZA system works on an uncomplicated basis. The input that is sent to the system is analyzed, the purpose of the analysis is to find keywords. In the case that keywords are detected, these sentences are converted and the conversion methods are associated with the rules based on those keywords. This process is executed for every keyword, that has not been transformed and retrieved earlier.

Text files are used to store rules for the ELIZA system. Each line in these text files must start with a command notation, these notations are used to define if the pattern is a welcome message, void input, key word action to be performed or others.. Different letters are used to represent cases of this notation, for welcome message the letter ″W″ is used. The letter ″Q″ is used to represent quitting messages. Void input is represented with the letter ″V″. Input transformations are marked with the letter ″I″ and key word patterns with the letter ″K″ and key word response patterns with ″N″. Output transformations are represented with ″O″. The script command notation for memorized phrases is "M". Comments are described with "", while actions to be performed start with "&". (B. Shawar & Atwell, 2002).

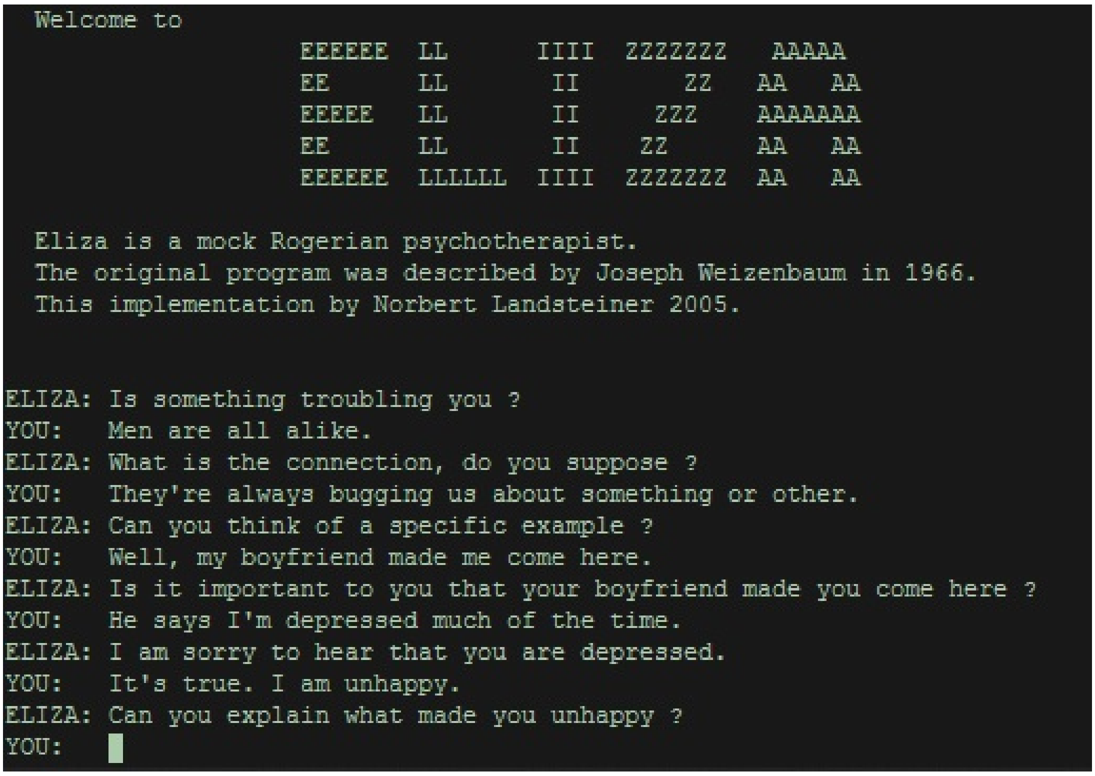

Figure 3.3.: Eliza Chatbot Screenshot (Wikimedia, 2018)

The ELIZA system begins its task of responding to the user by first searching for keywords in the input text. A keyword is a word that is defined in the ELIZA script and marked as relevant. The script attributes criterion numbers or ranks to keywords. These ranks or numbers are created by the programmer. In the case that a keyword is found, it is saved in the keystack. Keywords in the keystack are ordered descending by the highest rank. After this, the initial input is manipulated and transformed based on rules, that are associated with the highest keyword in the keystack. Words with a high rank are considered as high priority words in comparison with words that have a lower rank to conversations and are handled separately of contextual patterns (Shah et al., 2016). After the extraction of keywords and assignation of the correct rank to these keywords, it is necessary to find the correct transformation rule based on the found keywords. The transformation consists of two parts, the decomposition rule, and the reassembly rule. To find the correct disassembly rule for the keyword, different rules are applied until the correct pattern for the keyword is found. When the correct pattern is found the sentence is then disassembled using the rules of the keyword, it is also arranged into sections based on the same rule. The goal of the decomposition is to assign a specific reassembly rule. The reassembly rule is used to formulate a response. It uses the input and fragments of the input to generate a valid and acceptable response(Weizenbaum, 1966). These steps outline the majority of the methods which ELIZA pursues to produce a response from standard input. There exist specialized situations that ELIZA can reply to. One Weizenbaum explicitly addressed was the case when there is no keyword. One solution meant to have ELIZA respond with a statement that lacked content, such as "I understand" or "Go on.". The following method was to utilize a "MEMORY" structure, which memorized previous recent information and would use this information to generate a response referencing a section of the earlier conversation when faced with no keywords. This was feasible due to Slip's ability to mark words for additional usages, which simultaneously enabled ELIZA to analyze, save and reuse words for usage in outputs (Weizenbaum, 1966).

The code of the ELIZA system enables these methods of decomposition, inspection, and reassembly of data, but the specific methods are defined by the operating script of the chatbot. The operating scripts are editable, also new scripts may be created if a new operation is required. This also enables the application to be used in various conditions, including the popular DOCTOR script DOCTOR script, which simulates a Rogerian psychotherapist, but also a script called "STUDENT", which is able of using logical analysis parameters and applying it to give answers to problems of similar logic (A. U. Deshpande et al., 2017).

3.2.3. ALICE System

A.L.I.C.E. chatbot system, inspired by ELIZA, is an NLP chatbot system that applies pattern matching rules to converse with humans. A.L.I.C.E. like chatbot system stores their knowledge in AIML files. AIML stands for Artificial Intelligent Markup Language and it is derived from Extensible Markup Language (XML). AIML Files start with a AIML tag which carries the version information and contains data objects. These data objects are called AIML objects and are made up of sections named topics and categories. The topics and categories contain parsed or unparsed data. The topic is not a mandatory element, but if it is created it contains a set of categories that are related to the topic. The user input and the templates related to the response are contained in the categories. (Kerly et al., 2007).

AIML is an uncomplicated pattern and consists only of words and spaces. These words can also contain wildcard symbols. The wildcard symbols are _ and *. The words may consist of letters and numerals, but no other characters. The wildcards are considered to be words in the AIML pattern, it is important that words are separated by a single space. The pattern language is case invariant. Finding the best, longest, pattern match is the core of pattern matching techniques. AIML language consists of three categories the atomic Category, the default Category and the recursive Category. Atomic Categories are patterns that do not have wildcards _ or *. Listing 3.2 shows an example atomic category AIML code (AbuShawar & Atwell, 2015).

```
<category>
   <pattern>10 Liters of Water</pattern>
   <template>That is too much!</template>
</category>
```

Listing 3.2: AIML Atomic category example

Default categories are patterns with wildcard symbols or _ as seen in listing 3.3. The wildcard symbols can match any input.

```
<category>
   <pattern>Hello *</pattern>
   <template>Hey</template>
</category>
```

Listing 3.3: AIML Default category example

Recursive categories contain templates with <srai>and <sr >tags as seen in listing 3.4. The template keyword srai stands for simple recursive artificial intelligence and the template keyword sr stands for symbolic reduction. The applications of recursive categories are to divide and conquer by mapping various means of saying the identical thing to the corresponding reply.

```
<category><pattern>BINARY_REGISTER * 1</pattern>
  <template>
      <srai>BINARY_REGISTER</srai> 0
  </template>
</category>
<category><pattern>BINARY_REGISTER * *</pattern>
  <template><star/>
      <srai>BINARY_REGISTER</srai>
  </template>
</category>
```

Listing 3.4: AIML Recursive category example

To start the pattern matching process all inputs that go to the AIML interpreter must be normalized. (A. U. Deshpande et al., 2017).

3.2.4. Comparison of ALICE and ELIZA

Based on previous sections we can determine that there are differences in the two systems. A.L.I.C.E. systems are based on pattern matching algorithms and templates that represent input and inputs. While ELIZA systems use keywords and keywords patterns together with input and output rules to generate responses. Systems based on the A.L.I.C.E. system uses recursive techniques for pattern matching. This is considered one of the main features of the system. The rules in the ELIZA system are applied only once in order to disable infinite iterations of rules. The recursive nature of the A.L.I.C.E. system enables the separation of a sentence into multiple parts and then the combination of their result, this is not available in ELIZA systems. This is based on a depth-first search algorithm, which allows the system to find the longest matching pattern. In the ELIZA system, the response is generated according to the first keyword match. Both systems provide memorization for the past input and output for further usage. Eliza systems have to functionality called dynamic process, which allows multiple actions to happen while the conversation is in progress. By repeating the same input in a conversation, ELIZA systems try to provide different answers. The answers are randomly selected from a response list. Since A.L.I.C.E. systems do not randomly generate the response, similar or same responses are returned (B. Shawar & Atwell, 2002; A. U. Deshpande et al., 2017; AbuShawar & Atwell, 2015).

According to Bani & Singh (2017) A.L.I.C.E. system stored huge text corpora and ELIZA the system provides a grammatical analysis of sentences. It is easier to build an A.L.I.C.E. system since it uses simple patterns and templates together with pattern matching algorithms to match input and output. Usage of complex rules which need input and output transformations is a trait of an ELIZA system. Keyword patterns for user input representation are also necessary for an ELIZA system. It is possible to implement these representations by using simple pattern templates in an A.L.I.C.E. system.

3.2.5. Chatscript

Wilcox criticized several aspects of AIML and complained about the pattern only allowing exact words. In his view, the pattern matching should include the use of reusable synonym definitions. Due to these and other issues, Wilcox decided to develop an alternative to AIML by extending the AIML engine and created Chatscript (Bruce, 2008). The first ChatScript conversational agent that won the Loebner Prize Competition was Suzette in 2010. This was because Suzette managed to convince one of the judges that she was human. A more detailed explanation of the functionality of ChatScript can be found in chapter 6 where the framework is described as part of the practical section of this thesis.

3.3. Introduction to Dialog Systems

Dialog System or also known as conversation agent (CA) is a computer system that has the task of talking to a person and keeping the conversation structure coherent. Text, speech, gestures and other ways of communication are the input and output communication channels of dialog systems. In order to keep the conversation structure consistent, the usage of the dialog manager is a key feature for dialog systems. It follows the conversation and determines in which direction should it go (Dakkak et al., 2014). The main differences between dialog systems and chatbots according to Dakkak et al. (2014) are that dialog systems make use of more theoretically motivated techniques as seen in figure 3.4. They are also often developed for a particular field whereas chatbot systems aim at open domain conversation. Dialog Systems consist of input parsing (e.g. Speech to Text or just forwarding user text input to Natural Language Understanding, Natural Language Understanding and Natural Language Generation and Output Presentation. (Kluwer, 2011).

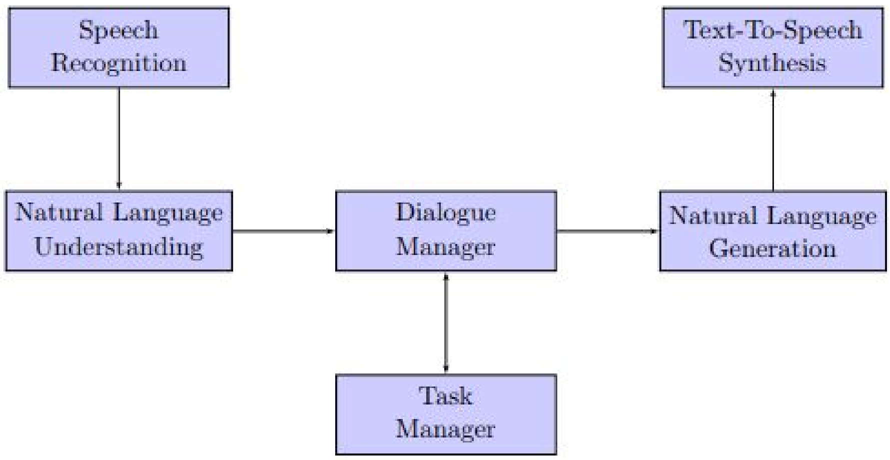

Figure 3.4.: Dialog System Structure according to van Woudenberg (2014)

These systems can have a global domain (e.g. Alexa) or a specific domain (e.g GPS navigation). Dialog systems have the capacity to track the state and the flow of the conversation. This is because they have a dialog manager. Dialog systems are hard to implement and can converse within some limited domain (van Woudenberg, 2014).

3.4. Chatbot Systems and Dialog Systems

Radziwill & Benton (2017) describe dialog system as a parent term, where Conversational Agents and Interactive Voice Response (IVR) systems are children terms of dialog systems as seen in figure 3.5. IVR systems are systems like telephone applications that respond to specific keywords.

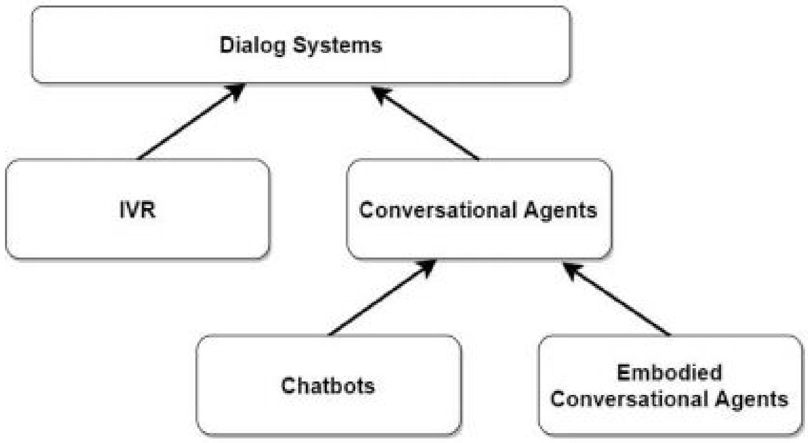

Figure 3.5.: The relationships of terms according to Radziwill & Benton (2017)

In most literature chatbots with NLP are considered dialog agents, but not dialog systems. Jurafsky & Martin (2018) classify conversational agents into goal-oriented conversational agents and chatbots. For business a goal-oriented chatbot is the most common chatbot, this chatbot is also called a transactions chatbot. These chatbots support tasks such as ordering products, retrieving information or checking product properties. For conversational chatbots, the focus lies in the conversation with the user, without understanding greatly what the user said. Conversational chatbots do not memorize the context of the conversation. One of the most common uses of a conversational chatbot is the replacement of classic FAQ pages. Pappu & Rudnicky (2014) defines dialog systems as goal-oriented knowledge acquisition strategies for knowledge acquisition. Answering simple questions and small talk can be easily implemented into a chatbot without the knowledge of linguistics and advanced language processing methods. But most conversations have more complexity and are hard to map into simple rules, that is why chatbots cannot fully understand conversations (van Woudenberg, 2014).

Conversational Agents can be classified according to three main criteria appearance, knowledge base, and initiative. Appearance is the way that they are presented to the user, the most simple representation of a chat agent is a text-only chat agent. A text-only chat agent means implies that user input, as well as the output, are both in textual form only.Bruce (2010) proposed the classification of CAs according to the type of knowledge that is utilized by a CA. Two types of CA can be distinguished, CAs that can retrieve their knowledge from a data-set of existing conversations, or a set of hand-crafted, predefined rules. The third category of CAs are CA systems according to who takes initiative in the conversation. The control of a conversation is defined by the initiatives taken in the conversation. (Radlinski & Craswell, 2017). Three types of initiatives in conversation can be recognized: (1) human control, where the user chooses the topic and direction of the conversation, (2) system control, the dialog is steered by the CA and (3) mixed-initiative, where each agent can contribute to the task what it does best (Horvitz, 1999).

3.5. Natural Language Processing

Natural language processing is an area of artificial intelligence that is focused on developing ways for machines to interpret and also to understand human language in natural form (Khurana et al., 2017). NLP has the goal to bridge the gap between human communication with the computer. It is a combination of many disciplines, including computer science and linguistics. A set of symbols which is organized by a set of rules can be defined as a Language. The symbols are combined and used to generate information. This combination of symbols is done according to the predefined rules. The NLP task is to analyze the symbols of the language while taking into consideration the rules of the language to extract information from sentences. NLP can be classified into two parts Natural Language Understanding and Natural Language Generation (Jurafsky & Martin, 2018).

3.5.1. Brief History of Natural Language Processing

The start of NLP can be traced back to the early nineteen-fifties, although earlier works can be also found. In the year 1954, the Georgetown experiment took place. The main goal of this experiment was the demonstration of a Russian-English machine translation system (John Hutchins, 2004). A decade later Joseph Weizenbaum wrote ELIZA. ELIZA was a simulation of a Rogerian psychotherapist (Weizenbaum, 1966). Although ELIZA had no information about human thoughts or emotion, the system was capable to provide human-like interaction. From the late 1980s, there has been a revolution in natural language processing with the advent of machine learning algorithms for speech processing. This was due both to the steady increase in computing power and to the gradual decline in the dominance of Chomsky's theories of linguistics (e.g. transformational grammar), whose theoretical foundations discouraged the kind of corpus linguistics underlying the machine learning approach lies speech processing (Kates, 1976).

3.5.2. Natural Language Processing Applications

Natural language processing is used in a lot of everyday technologies. Some of the applications are based on grammar correction or spelling correction, these applications can be found in all text editors. Besides these functionalities, NLP can be used in fields like automatic summarization, dialog systems, sentiment analysis, text classification, question answering and more (Jurafsky & Martin, 2018). One of the most commonly used frameworks for natural language processing is Natural Language Toolkit (NLTK). NLTK was created in 2001 in the Department of Computer and Information Science at the University of Pennsylvania (Bird et al., 2009). Steven Bird & Loper (2009) describes the architecture of a simple natural language based dialogue system as shown in figure 3.6.

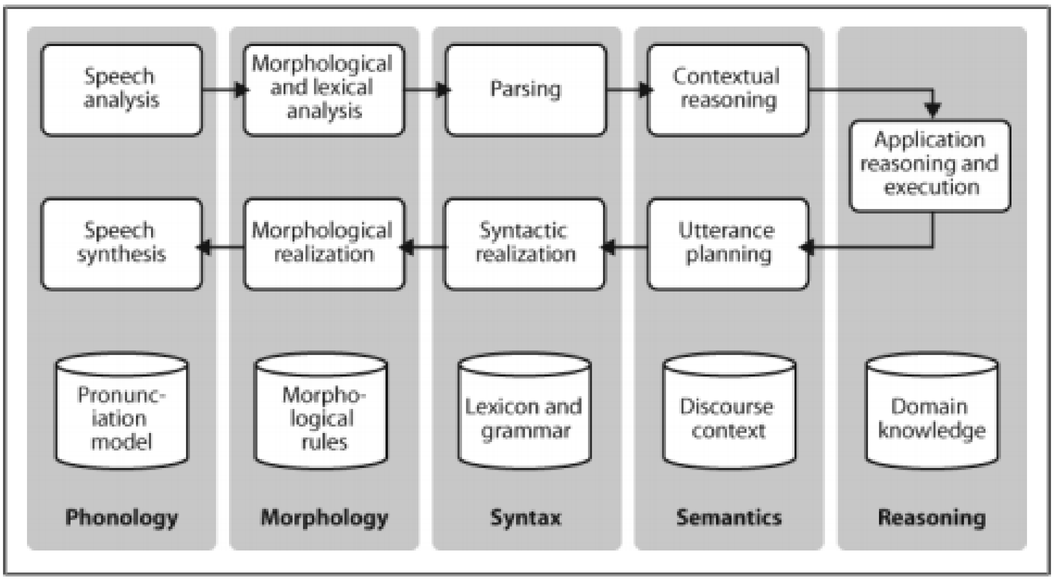

Figure 3.6.: Simple pipeline architecture for a spoken dialogue system according to Steven Bird & Loper (2009)

At the top of the figure, from left to right, is a pipeline of a language understanding components. It includes speech input via syntactic parsing to meaning representation.

Along the center, from right to left, is the pipeline of components for converting concepts to speech. At the bottom is static information: the repositories of language-related data that the processing components require.

3.6. Machine Learning

Machine Learning (ML) can be formally defined as a scientific study of algorithms and statistical models used by computer systems to efficiently perform an assigned task based on patterns and inferences without being explicitly instructed on how to do that task (Bishop, 2006).

3.6.1. Brief History of Machine Learning

The history of machine learning starts with the concepts of statistics. The fundamentals of machine learning were created in the years before 1940 in the work of Thomas Bayes, Pierre-Simon Laplace, Adrien-Marie Legendre, Andrey Markov, and many others. In the late 1940s the first manually operated computer system, ENIAC, was invented (McCartney, 1999). At the time ENIAC was invented a person with intensive numerical computation capabilities was called a computer, so the ENIAC was initially called a numerical computing machine. Which sparked the idea of building a machine that was able to emulate the human way of thinking and learning. Arthur Samuel developed a computer program in the 1950s which was able to play checkers by using simple learning and memorization techniques. He was also the first one who came up with the phrase "Machine Learning" in the year 1952 (Wiederhold & McCarthy, 1992). Later, Frank Rosenblatt combined the ideas of Donald Hebb's model of a brain cell and Arthur Samuel's machine learning breakthrough to create the perceptron Rosenblatt (1958). The Perceptron was described as the first successful neuro-computer. In the later years, machine learning was improved with various algorithms and structures, such as the nearest neighbor algorithm, and artificial neural networks. These advances made it possible to reach the current state of machine learning, where it is used in almost every sphere of life, such as analysis of sales data, fraud detection, product recommendation, dynamic pricing, and NLP.

3.6.2. Popular Machine Learning Approaches

Machine learning approaches are organized based on the type of learning. These algorithms can be described with four different types of learning supervised machine learning algorithms, unsupervised machine learning algorithms ,semi-supervised machine learning algorithms and reinforcement machine learning algorithms. Supervised learning algorithms are described by a function that maps directly input values to desired output values. One standard supervised learning task is a classification problem. Unsupervised learning algorithms create structure or patterns for a specific dataset without a defined output variable (Shalev-Shwartz & Ben-David, 2014). Semi-supervised machine learning algorithms combine both functionalities of supervised and unsupervised learning approaches in order to create a function or a classifier for a specific problem. Reinforcement learning problems involve learning about taking suitable action to maximize a numerical reward signal in a certain situation (Sutton & Barto, 1998).

3.6.3. Conversational Agents and Machine Learning

Dialogue Generation or Intelligent Conversational Agent development with the use of machine learning techniques is an intriguing problem in the field of Natural Language Processing. ELIZA, A.L.I.C.E, and ChatScript are rule based, there are also CAs that use ML algorithms to extract information from data in order to understand and reply to user sentences. Machine Learning algorithms try to find estimate behavioral patterns in training data sets by identifying patterns in the data(Tiha, 2018). With the appearance of advanced machine learning algorithms, CAs have also adopted modern methods of development. Csaky (2019) used Deep Learning approach to train and develop a CA. Since these machine learning algorithms are based on large amounts of data to create a CA, they can also be classified as data-driven approaches for CA development.

3.7. Summary

Chatbots have been evolving since the early 1960s using simple pattern matching techniques to answer questions to machine learning methods. ELIZA and ALICE are the most influential chatbot systems that have shaped and inspired the creation of modern chatbot systems and methods. This development of different methods was accelerated by the introduction of the standardized Turing test and the Loebner Prize. The Turing test questions the ability of the machine to answer questions as a human and its display of intelligence. The Loebner Prize is awarded to chatbots who are considered the most human-like.

Nowadays, chatbots do not only answer simple user questions but have the goal to lead a conversation with the user. Chatbots that are capable to hold a conversation are classified as conversation agents. The connection between chatbots and dialog systems has been evaluated and the term conversational agent has been selected as the umbrella term that covers both chatbots and dialog systems. The availability of large conversations data sets and improvement of machine learning approaches led to increased interest in data-driven approaches for CA development. Data-driven approaches use machine learning algorithms to learn from large data sets to create conversation models. Natural language processing has been defined as the link between data-driven approaches and conversational agents. Natural language processing is an area of artificial intelligence that is focused on developing ways for machines to interpret and also to understand human language in natural form. The main focus is on developing methods that aid computers with the process of analysis of large amounts of natural language data. Natural language processing can be used in fields like automatic summarization, dialog systems, sentiment analysis, text classification, question answering and more.

Different ways that CAs interact with the user in a conversation are: human control, where the user chooses the topic and direction of the conversation, system control, the dialog is steered by the CA and mixed initiative, where each agent can contribute to the task what it does best.

4. Conversational Agents in Customer Service and Information Retrieval

This chapter will go through the appearances of conversational agents in modern customer support systems. Benefits and the downsides of CAs in customer service will be defined together with modern platforms and applications that enable the usage of CAs in customer support. The focus of the chapter is the definition of a CA as an additional communication channel to an already existing number of channels in the area of customer support systems.

4.1. Conversational Agents Platforms

S Graphical User Interfaces (GUI) have been the main way the users interact with machines since the 1980s. The shrinkage of the size of gadgets like laptops and mobile phones have reduced the screen sizes and increased the need for user interfaces to become invisible or at least to also shrink in size. This fact in combination with the recent development in the area of Artificial intelligence has increased the popularity of conversational user interfaces(Janarthanam, 2017). Conversational agent platforms can be classified in many different categories, but the main tree families among the existing platforms are no programming platforms, conversation-oriented platforms and platforms backed by technological giants (Couto, 2017).

4.1.1. No Programming Platforms

No programming platforms are non-technical user-oriented platforms.These platforms focus on providing easy to create chatbots without much programming included in the process. This implies that NLP and Machine learning methods can not be easily integrated into these platforms. With these type of platforms, you can develop a simple chatbot very quickly and the learning curve is low. Chatfuel, Massively, ManyChat, and MobileMonkey are just some of the available platforms in this area. Even though the learning curve is low, the user interface of these platforms is not so easy to understand as seen in figure 4.1. Chatbots build through these platforms become unusable when the chatbot logic increases in complexity, this is because advance techniques like NLP are not implemented. (Janarthanam, 2017; Couto, 2017).

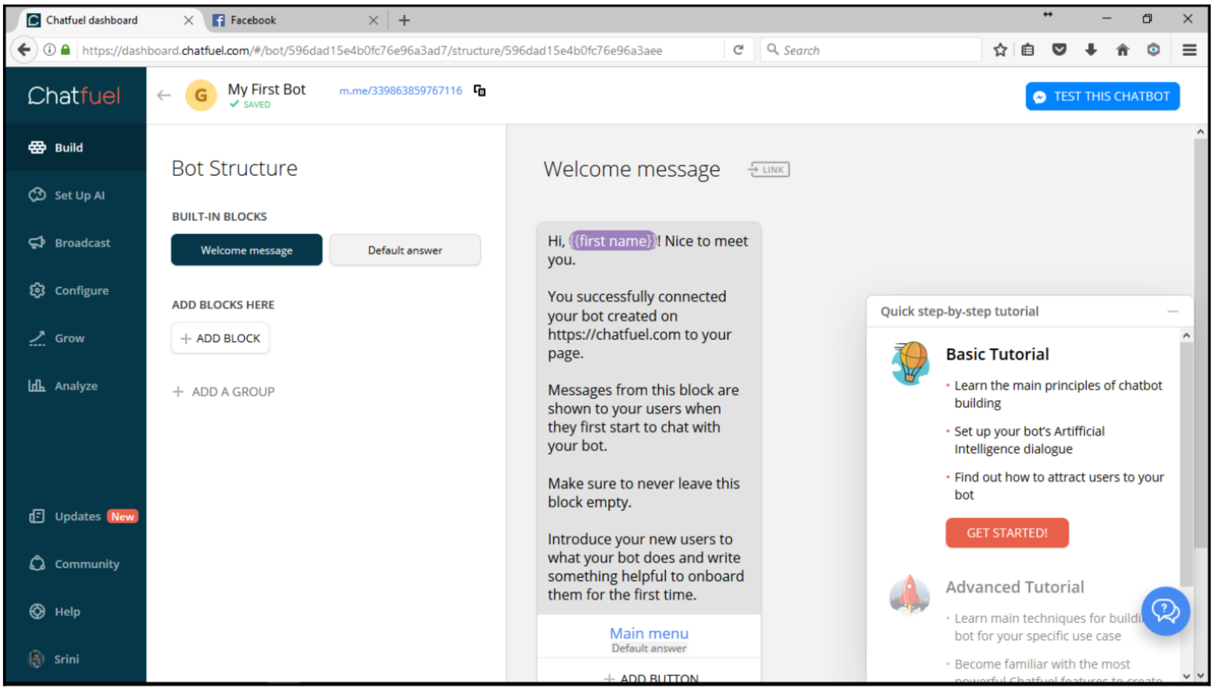

Figure 4.1.: Chatfuel User Interface Janarthanam (2017)

4.1.2. Conversation Oriented Platforms

Retrieving questions to answers and retrieving information is possible with the use of chatbots that were created through these platforms, but the focus of these platforms is to enable the creation of chatbots that can converse with users. Specification languages like AIML are used to program dialogs on these platforms. They are very flexible to create complex conversations. Scaling these systems would prove to be difficult if the pattern creation would be done manually. (Valtolina et al., 2018).

4.1.3. Platforms Backed by Technology Giants

Tech companies as google[1], facebook[2] and amazon[3] developed these platforms. Platforms from this category are Dialogflow (Google), Wit.ai (Facebook), LUIS (Microsoft[4]), Watson (IBM[5]) and Lex (Amazon). These platforms have a specification language and a graphical user interface, which enable the users of these platforms to develop complex

[1]www.google.com
[2]www.facebook.com
[3]www.amazon.com
[4]www.microsoft.com
[5]www.ibm.com

chatbots. Since the coding is done by combining a programming language and user interface, the learning curve for these platforms is low. The negative side of these platforms is the cost, due to the high performance and usability, the price for creating a chatbot is very high.

Table 4.1 summarizes the benefits and concerns of different platforms used for development of conversational agents.

Platform	**Benefits**	**Concerns**
No Programming Platforms	Low learning curve. Fast and easy development.	Not usable for complex tasks.
Conversation Oriented Platforms	Well defined samples and starter code. Chatbots build on these platforms are able to solve complex tasks.	High learning curve.
Platforms Backed by Technology Giants	Learning curve for these platforms is low. Well defined samples and starter code.	Price to operate chatbots on these platforms is high. Cannot be integrated into existing systems.

Table 4.1.: Benefits and Concerns of Conversational Agents Platforms

4.2. Applications

Based on chapters 1 and 2, this section will focus on the description of applications that implement two functionalities of conversational agents. One functionality is a replacement for customer support agents with a conversation agent, whose main goal is to assist the customer. The second role is the role of an information retrieval system. Here a conversational agent is used as a replacement of an existing web application that retrieves information from a knowledge base.

Table 4.2 describes different conversation agents that have been build in order to serve as information retrieval or customer support systems. The example conversation agents have been implemented as either standalone applications or extensions to existing systems.

System	Description
DocChat	DocChat can respond to utterances by using information retrieval approaches on unstructured documents. (Yan et al., 2016).
ManyChat	Representation of a real person helping the user get content from a web page, implemented as a facebook messenger bot. (Molly, 2017).
FAQchat	ALICE based chatbot used for answering frequently asked questions (B. A. Shawar et al., 2005).
YPA	The retrieval of information from the British Telecom's Yellow pages is the functionality of the YPA dialog system implemented in natural language (Abu Shawar & Atwell, 2007).
Virtual Assistant Leo	Chatbot system that improves communication and reduce efforts required for providing information to the existing and potential students (Mislevics et al., 2018).
SuperAgent	The SuperAgent chatbot uses a large-scale e-commerce database which is publicly available as its knowledge base. (Cui et al., 2017).
HC (Hotel Chatbot)	Conversation Agent that simulates an intelligent conversation with Web page visitors, dedicated to hotels. The benefit of the HC is that it can provide more information to the user than any other web sources available. It also provides useful guidance when he is navigating the hotel website and does promotion of the hotel and the surrounding of the hotel. (Lasek & Jessa, 2013).

Table 4.2.: Conversation Agent Examples for Information Retrieval and Customer Support

4.2.1. Conversational Agents in use for Customer Support

As mentioned in chapter 1, conversational agents have found usages in customer support. The ability to answer a huge number of user requests in a short time with high accuracy makes them extremely useful. The use of chatbot platforms backed by technology giants to implement conversational agents for customer support has become a standard. this is due to the low learning curve and availability of these platforms.

The main role of these chatbots is to engage with the customer and try to answer questions. One of these chatbots is ManyChat. ManyChat is a representation of a real person helping the user get content from a web page (Molly, 2017). Additional usages of chatbots in customer support, are replacements for frequently asked question pages. In this case, chatbots give simulate to the user that he is talking to a real person instead of browsing a static web page (R. Ranoliya et al., 2017). Additional benefits of using conversational agents as replacement for customer support agents is that the users do not experience discomfort when asking the same question multiple times (Initiative, 2017).

4.2.2. Conversational Agents in use for Information Retrieval

Savoy & Gaussier (2010) describes information retrieval as the process of obtaining information that is relevant to an information need. Information retrieval is the ability to search for information in different multimedia sources like texts, images, videos, or sounds.. Web pages are user interfaces that enable the user the ability to retrieve information from a knowledge base. Browsing is a method of information retrieval, where the user navigates through a web page to find the necessary information (Abdulghani et al., 2018).

4.3. Summary

Graphical User Interfaces have been changing since the 1980s. With the reduced size of consumer devices like mobile phones and laptops, the need to have a compact interface made conversational user interfaces popular. The popularity of conversational agents in the business fields have sparked the creation of frameworks and platforms that aim to enable fast creation of conversational agents. Most common platforms that are used for the creation of conversational agents are no programming platforms, conversation oriented platforms and platforms backed by technology giants. No programming platforms are used by individuals without programming experience to create simple conversation agents. Conversation oriented platforms are sophisticated platforms that leverage AIML or other frameworks for the creation of complex chatbots. These platforms are aimed at individuals who have formal education in the programming field and have a good understanding of conversation agents. Platforms backed by technology giants are a combination of no programming and conversation oriented platforms. Creation of complex chatbots with no or low amount of programming knowledge is a key feature of these platforms. These platforms have been used to create a wast variety of conversational agents like DocChat, ManyChat, FAQChat, YPA, Leo, SuperAgent and more.

There are many use cases of conversational agents, but the ones that are commonly implemented are replacements for customer support agents and systems for information retrieval. As replacements for customer support agents, chatbots have to goal to converse with the customers and answer frequently asked questions. Chatbots as information retrieval systems reduce the time and effort a user needs to retrieve information the classical way through a web application/page or an information retrieval system.

5. System Architecture and Specifications

Based on chapters 1 and 2 where the motivation from the companie's side was presented, functional and non-functional requirements will be defined. These requirements together with the findings from chapters 3 and 4 will be used for the technology selection and creation of the system architecture.

The goal and objective of this thesis is a working chatbot system demonstrator that uses existing technologies and/or improves the communication channels of the company. The prototype system will focus on retrieving user information and answering frequently asked questions.

5.1. Analysis of the Current System

To formulate a conceptual architecture two main aspects for the current system have been selected for the analysis: functionality and technology. The functionality aspect covered the functional features of the system, what functions were offered to the user and how these were accessible. To plan the prototype in a way that it can be integrated into the existing system, the technological aspect analysed frameworks, programming languages, and libraries that were used in the current system.

5.1.1. Functionality

The company smaXtec animal care GmbH provides a software package that acts like a herd monitoring service. This software package gives the user access to valuable information related to their animals. This information includes heat, calving and health alarms, and notifications. Figure 5.1 describes the components of the smaXtec Inside Monitoring System. This system enables the user to view information about the cows on a mobile device instead of constantly visiting the animal. The main use case is the retrieval of animal information via a mobile device. This retrieval is done with the use of the smaXtec messenger application[2]. Due to the limited technical capabilities of the

[2]https://messenger.smaxtec.com/

average user of the smaXtec system, this system is designed to be easily understood and all the functionalities have been simplified.

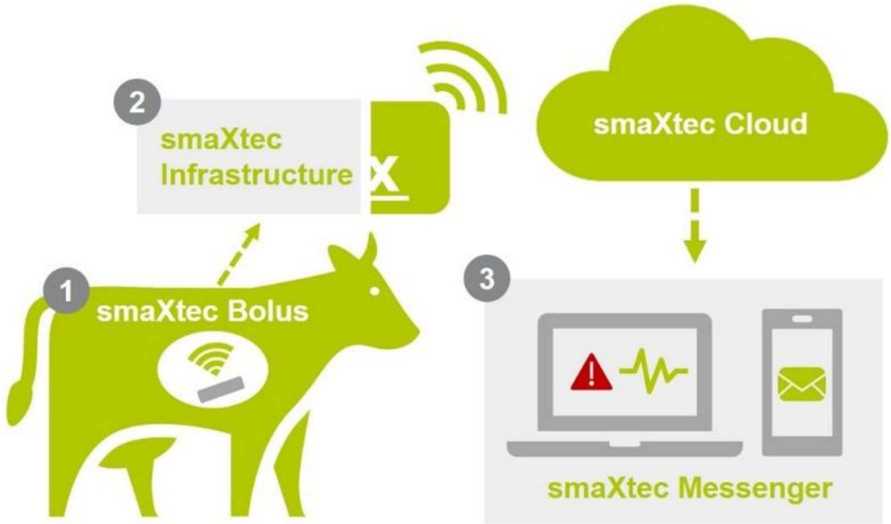

Figure 5.1.: Components of the smaXtec Inside Monitoring System Hogeveen et al. (2017)

Figure 5.2 visually show basic animal data with the usage of graphs. The user can see temperature and activity data of a cow, together with important cow events that occurred. The focus of the application is on showing cow events to the user.

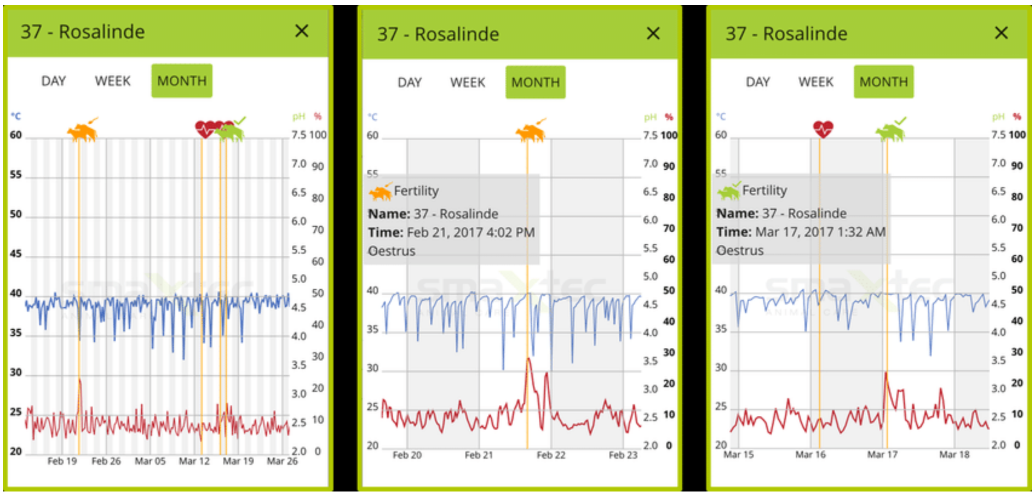

Figure 5.2.: Graph of two heat events of cow Rosalinde. Hogeveen et al. (2017)

During the years of the existence of the company different questions have been repeatedly asked. This has made it necessary to create a web site with frequently asked questions seen in figure 5.3. The purpose of this page is to help current and future users of the smaXtec system to get information regarding the system. According to the testimonials of the employees, most questions that the users ask the support team can be answered by looking up the FAQ page on the web site. It is the fact that the customers prefer to talk to a representative then to research the FAQ page.

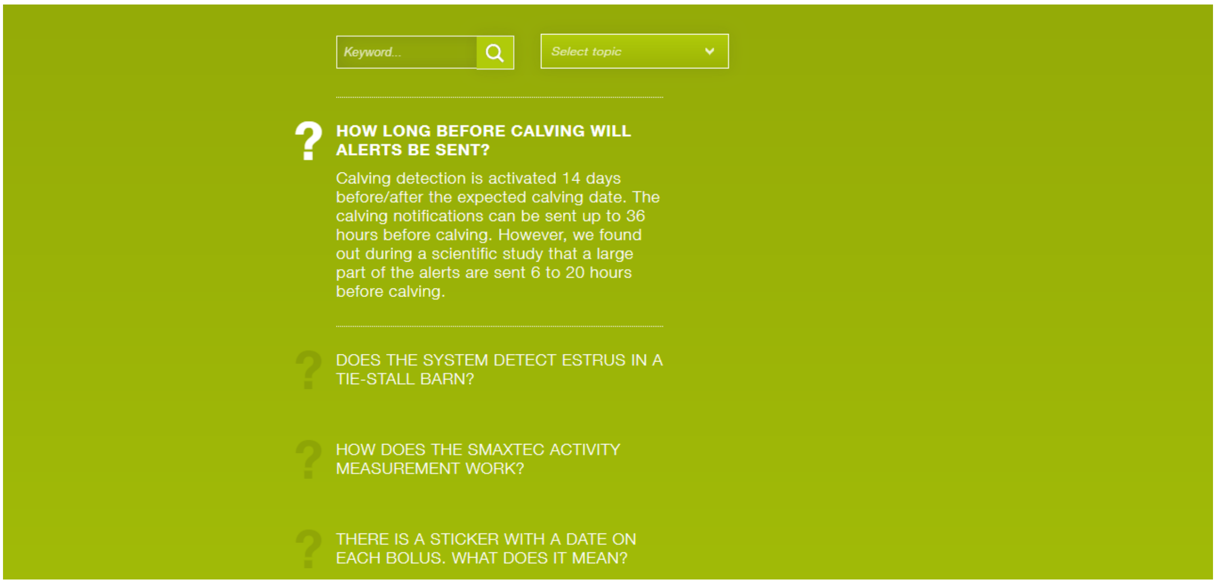

Figure 5.3.: Frequently Asked Questions Web Site [1]

5.1.2. Technology

The platform was build in a server-client architecture. The backend applications were developed using the Python programming language and the Flask framework. The Flask framework enforces a Representational State Transfer (REST) interface for accessing different functions. Besides the backend applications, the data processing layer is also implemented using the Python programming language, the main reason for this is the performance and speed of Python.

The web interface and the mobile applications use the REST interface to send and retrieve and send data from/to the backend system. This architecture allowed to reuse functionality independent of the UI implementation. The web application was hosted on an external server using the Google Cloud Architecture, using a Linux operating system. The web application interface uses HyperText Markup Language (HTML), Cascading Style Sheets (CSS), and JavaScript (AngularJs). Calls to the server were made using asynchronous JavaScript and XML (AJAX). For the mobile applications, different technologies were used. Mobile applications were available for Android and iOS. Both applications use React Native as the programming language. React Native is a native code wrapper that includes HTML, CSS and JavaScript interface.

Figure 5.4 describes the architecture of the smaXtec System. Where the web interface is grouped into the front end component.

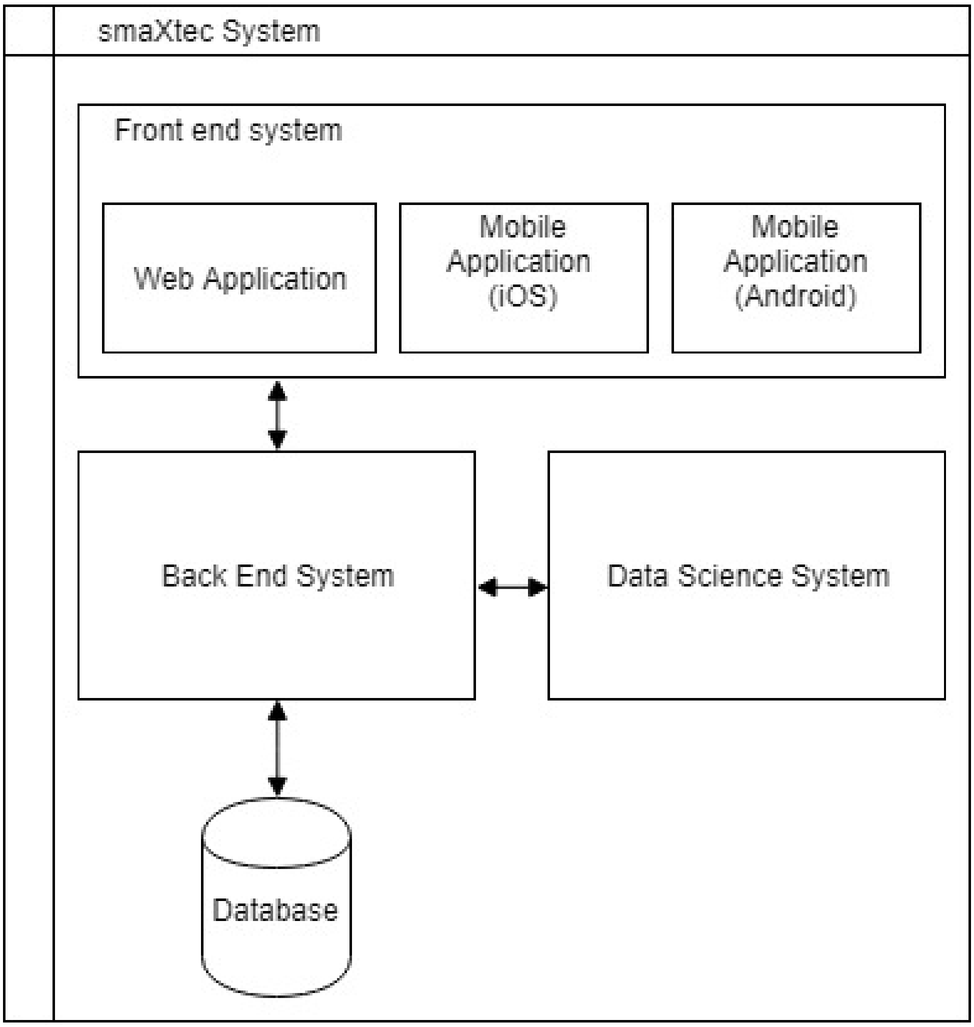

Figure 5.4.: smaXtec System Architecture

5.2. Requirements

To formulate the conceptual prototype idea and to create the system architecture it is necessary to assert the main requirements of the system. Functional requirements are those that relate to the technical functionality of the system. A non-functional requirement is a requirement that specifies criteria by which system performance can be assessed under certain conditions, rather than a particular behavior (Dabbagh et al., 2015). Both functional and non-functional requirements are stated below to determine and design the architecture of the software solution.

5.2.1. Functional Requirements

As stated in the section before, functional requirements describe the technical functionality of the system. Based on the motivation section stated in chapter 2 the functional requirements are are:

- The chatbot system must respond when a conversation is initiated
- The chatbot system must work on a web client
- The chatbot system must work on a mobile device
- The chatbot system should retrieve information about animals for the user via a natural language interface
- The chatbot system should provide answers to frequently asked questions via a natural language interface
- Frequently asked questions are asked in natural language form
- Information retrieval queries are formulated in natural language form
- The system can understand questions and queries only in the English language
- The chatbot system should be able to query data from the smaXtec backend system
- The system is capable of saving unanswered questions and respond to the user that an answer does not exists
- Administrators of the chatbot system can extend the system by answering not answered questions or adding question and answer pairs to fill the knowledge base

As stated in chapter 2 the CA should be able to answer questions about user data (e.g. ”How many cows do I have?”) and questions related to the company (e.g. ”What is a smaXtec Bolus?”). User interaction should be via a chat interface in natural language in the form of text. The input would be analyzed to understand the meaning. To improve the performance of the CA, the number of questions the CA can answer should be extendable. This means that the chatbot should be able to learn how to answer new questions.

5.2.2. Non-Functional Requirements

After the analysis of the smaXtec system, non-functional requirements were extracted. Previously mentioned in chapters 1 and 2 the CA system should serve as an extension to already existing communication channels and as a information retrieval system. This in terms means that the CA should have a user interface that is implemented in a HyperText Markup Language (HTML), Cascading Style Sheets (CSS), and JavaScript-based language. Which will enable easy integration into the existing smaXtec frontend system. The backend and processing parts of the CA have to be implemented in python to fit the needs of the company since all employees are skilled in python and use it as their primary programming language. The communication between the frontend and backend system should be over Application Programming Interface (API) calls. These calls have to be made using asynchronous JavaScript and XML (AJAX). The CA system has to run on a Linux system because company systems run on Linux based machines. One important aspect of the development is that the whole CA system should be run in-house. This means that no external CA or cloud hosting systems can be used. The CA should be able to seamlessly and without interruption communicate with the user. The availability of the chatbot is not a high concern, but the system has to be implemented in a way that it is scalable and redundant to provide a high level of availability. The CA should reply to the user on time to retain the conversation progressing without interference. In the case that there is no answer to a question from the user, the CA should notify the user about it in an understandable way. It must perform consistently in user-acceptable behavior when communicating.

5.3. Prototype Concept

After the analysis of the current smaXtec system, knowing the features the chatbot should support, describing different types of CAs, technical requirements can be extracted and appropriate technology chosen for the implementation. Based on the previously mentioned two prototypes were suggested.

5.3.1. Concept Proposal 1: Retroactive CA for user update

The first prototype concept proposal was a CA that provides interactive alerts to the user in a natural language. This would motivate the user to interact more with the system and the chatbot. Although the idea for this concept sounded promising, it did not fulfill all requirements. The time effort that was necessary to implement this prototype was out of scope, because of the technical limits. The notification system that was necessary to deliver these messages to the user, does not have the necessary capabilities.

5.3.2. Concept Proposal 2: CA for user data retrieval and question answering

The second concept was a CA that has the ability to understand natural language and provide answers based on a knowledge base. This knowledge base is build up by the company employees and maintained. Thus this concept needs a followup administration panel in order to enable daily improvements to the CA. This concept also fulfills all needs that were stated in the requirements. The CA will be able to evolve over time and answer more questions. The learning process of the CA will be semi-supervised.

Based on the initial motivation, requirements and technological restrictions concept proposal 2 was chosen to be implemented.

5.4. Conceptual Architecture

Based on the selected prototype proposal and the functional and non-functional requirements a conceptual architecture, shown in figure 5.5 has been suggested.

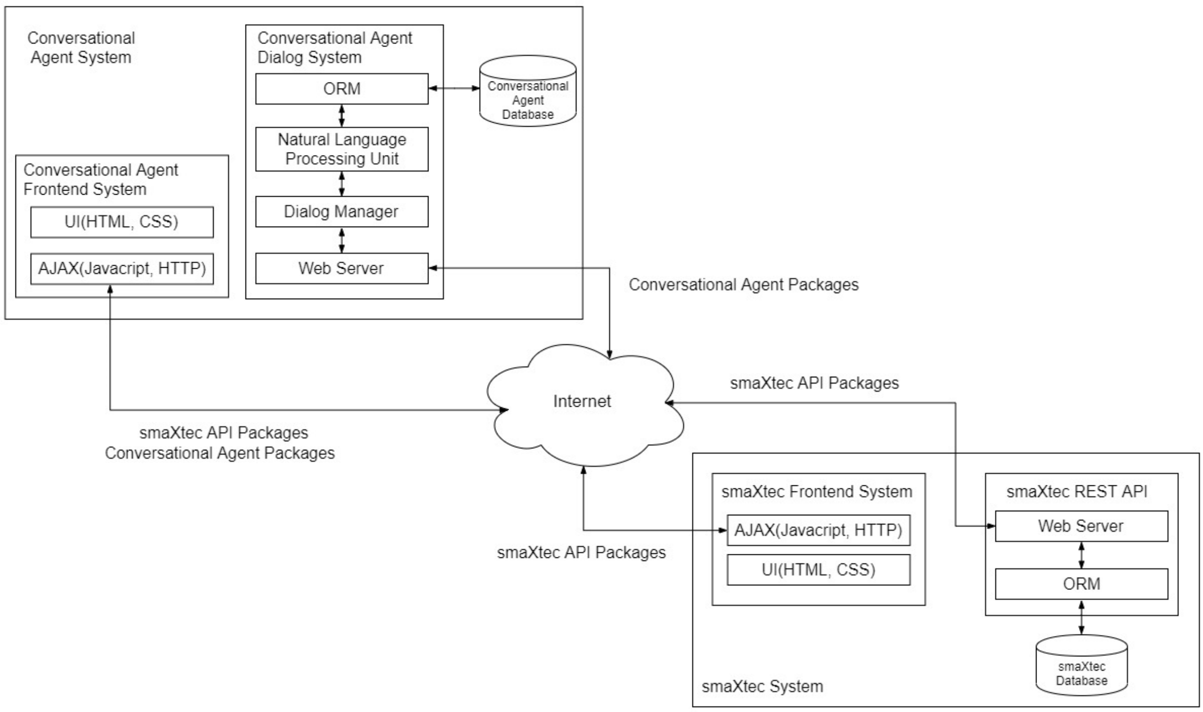

Figure 5.5.: Conversational Agent System Conceptual System Architecture

The Conversational Agent System has been split up into two main components on the conceptual architecture, the conversational agent frontend system and conversational agent dialog system. Based on the requirements, the conversational agent system has to have a stand-alone database. The mentioned main components have been defined as separate units because of the functionalities that they fulfill. The conversational agent frontend system is used as a presentation layer for the chatbot where the user interface has to be implemented in HTML and CSS and the communication layer of this component uses AJAX HTTP requests to communicate over the internet with the internal smaXtec system and the conversational agent dialog system. The tasks of natural language processing, managing the dialog and preserving user data and question-answer pairs are the main obligations of the conversational agent dialog system. This system also communicates with the smaXtec Rest API via the internet using HTTP requests. Two main messages and data packages are sent via the internet between these two systems. Conversational agent packages are responses generated via the conversational agent system and these are user queries in natural language and textual data that have been generated via the conversational agent system as answers to user queries. The other type of packages that are sent are the smaXtec API packages and these contain customer-related data. Animal data, user-specific data, and farm data are considered customer-related data.

5.5. Technology Decisions

To create a CA that can fulfill all the requirements stated in the previous sections, it was necessary to find a good CA framework that is available. Three CA frameworks were selected: AIML, dialogflow, and ChatScript. These frameworks were evaluated on a couple of factors and described in sections 3.2.3 and 3.2.5. The selected framework needs to provide an easy way to extend the existing CA. This means that the framework needs clear and easy to understand documentation. The framework should be open source since it had to be hosted in-house. Due to the nature of the system, that it needs to gather data about the customers, privacy was an important aspect of the framework. The framework needed to be performant due to a large number of customers. Decisions about the logic and communication components of the implementation and the technology used for the user interface are discussed in this section.

5.5.1. AIML

Introduced in Chapter 3, AIML is an XML based language that is used to create chatbot interaction rules. It enables extensibility, but the number of rules needed for a simple conversation is large. This framework provides good developer documentation and it is possible to host in on an own server.

5.5.2. DialogFlow

DialogFlow is a platform backed by a Technology Giant, this company is Google. DialogFlow created chatbots works based on intents and contexts. The intent is defined as a way to connect user messages to actions that should be executed by the Chat Agent. The context is used to determine correct intents based on previous knowledge. When DialogFlow receives a user request, it is first trying to classify it to determine if it matches a known intent.

5.5.3. ChatScript

ChatScript is a combination NLP engine and dialog management system designed initially for creating chatbots. But it has also found its use in different Natural Language Processing areas, because of its powerful Natural Language Processing engine. Bruce (2008) states that the flaws of AIML, that it requires a large number of rules in order to implement simple conversation. ChatScript can be considered as a improvement to the already existing AIML, in the form that it reduces the number of rules. According to Bruce (n.d.) the A.L.I.C.E. system needed around 120,000 rules to lead a conversation. This number can be greatly reduced with the usage of Chatscript.

5.5.4. Framework Selection

In order to select a suitable framework, key features for the selection have been extracted. They are In-House Hosting, Data Privacy, Documentation and Extendibility. Besides these CA frameworks, others have been analyzed. These frameworks either fit in chatbot platforms that are backed by technological giants or no programming platforms as explained in section 4.1. Which do not fulfill the requirements of data privacy and in-house hosting.

All frameworks mentioned in this chapter enable the creation of a flexible and extendable CA, but not all fulfill all the necessary requirements. Based on the characteristics described in Table 5.1 it is visible that Chatscript fulfills all the necessary requirements better than the other frameworks.

Based on the conceptual architecture from figure 5.5 it is visible that the "Conversational Agent Dialog System" needs to handle HTTP requests, connect to a database and act as a web server besides the ability to act as a natural language processing unit and dialog manager. The frameworks that were mentioned above do not have that capability, so an extension to the "Conversational Agent Dialog System" had to be chosen. To extend it with a web server, a python based web server had to be chosen due to the requirements of the company. The web server part of the "Conversational

Framework	In-House	Data Privacy	Documentation	Extendibility
AIML	Yes	Since it can be hosted in-house it offers a high level of data privacy.	Documentation in the form of blog articles and specialized documentation pages	It is easy to extend rules, but the number of needed rules increases exponentially.
DialogFlow	No	The data has to be hosted on the Google platform	Documentation in the form of blog articles.	It is easy to extend rules and apply machine learning to the existing rules
ChatScript	Yes	Since it can be hosted in-house it offers a high level of data privacy.	Documentation in the form of blog articles and specialized documentation pages	It is easy to extend rules.

Table 5.1.: Characteristics of selected CAs

Agent Dialog System" will be the backend part of the CA System. Because the backend component does data pre-processing, that is reading, saving and forwarding the user query to the ChatScript component, this component was named the "PreProcessing Server"

5.5.5. PreProcessing Server

To fulfill the previously mentioned requirements and the standards of the company, it was decided that the backend and logic layer should be implemented in the python programming language. Flask is the preferred python framework used in the smaXtec system, which is why it was selected to be the framework for the backend and logic layer. All operations that cannot be executed within the ChatScript framework should be executed there. The communication of different components in the smaXtec system was made over Hypertext Transfer Protocol (HTTP). According to the requirements, the backend layer should be able to save user queries and question-answer pairs into a database.

The backend part of the "Conversational Agent Dialog System" has to be able to communicate in the fastest possible way with the ChatScript framework. WebSocets protocol has been chosen as the preferred protocol to enable communication between the Backend and ChatScript Framework, because the two components can be connected all the time and share information contrary to the HTTP protocol, where each request has to start with a new connection between these two components.

5.5.6. Database

The internal database of the "Conversational Agent System" has to have the ability to store large textual data since it will store questions, answers and conversational data in the textual form together with relationships between these entities. There are many relationship database management system, but only a few of them work well with large textual data. Le et al. (2015) describes the advantages of PostgreSQL over other relational database management systems for storing and retrieving large textual data.

5.5.7. Conversational Agent Frontend System

The CA should be available to the user via a mobile phone or desktop PC as a standalone component or as an integrated component in already existing smaXtec applications. Already existing smaXtec frontend applications are based on HTML, CSS, and JavaScript (AnagularJS), consequently, the implementation of the frontend application should be done with the same technologies. The CA frontend should contain a chat and an administration panel application. The communication with other system components should be done via HTTP requests using AJAX. The chat application should resemble modern chat application like Facebook Messenger or WhatsUp, to lower the time that the user needs to learn how to use the application.

5.5.8. Integration and Communication

Besides the technological aspects of the application, it is also important to decide on how the CA system should be integrated into the system. It was decided to implement the CA system as a separate project, so it does not interfere with the operation of the existing smaXtec system. This in terms means that the CA system should have a frontend component that can be integrated into existing smaXtec frontend applications, while the backend logic runs on a server that is separated from other smaXtec applications.

It was decided that the communication between the CA system and the smaXtec system should be via HTTP, since the smaXtec system already offers a REST API. The messages sent will be in JSON representation format, since JSON is the standard for REST API communication over HTTP.

5.6. System Architecture

In chapter 2 the architecture of the smaXtec system is explained and section 5.4 provided the conceptual architecture based on requirements and prototype suggestion. Based on these and the selection of technologies for the components of the conceptual architecture this section describes the system architecture described in figure 5.6. Two parts of the smaXtec system are crucial to the architecture of the CA. These parts are the smaXtec backend and smaXtec frontend system. For the CA system to provide an extension to the already existing system, it has to provide the ability to retrieve information to the user and to get answers to specific frequently asked questions. This is achieved by connecting the CA system to the smaXtec backend system via the smaXtec REST API and creating an internal database for question-answer pairs. The conceptual architecture schema for the chatscript and backend logic can be seen in figure 5.6.

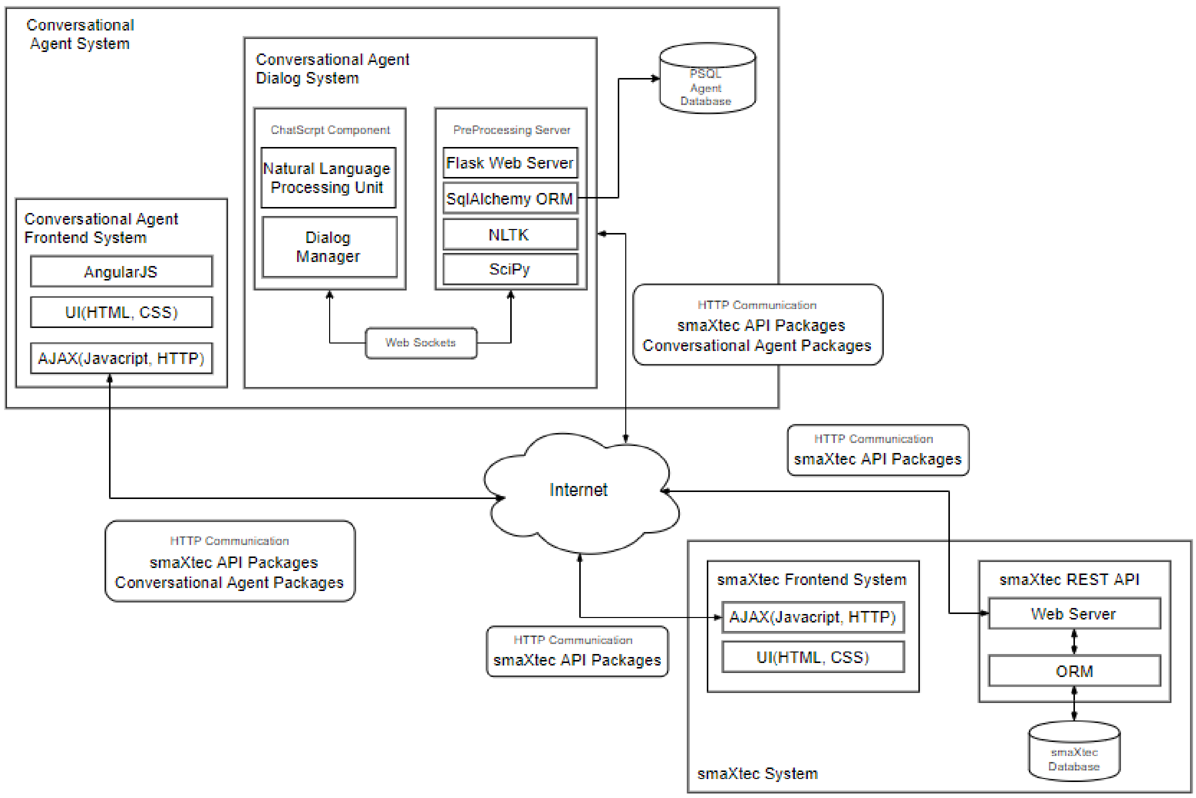

Figure 5.6.: Conversational Agent System Suggested System Architecture

Since the ChatScript framework operates as a standalone component, it can be physical separated from the Preprocessing Server. This enables the "Conversational Agent

System" to be scalable if the number of requests to one of these components reaches a level where one instance cannot handle these requests, which was mentioned as a requirements from the company's side.

All question-answer pair data can be stored in a centralised database and can be accessible from all preprocessing server instances. The preprocessing server is used to handle HTTP requests from the user. The Preprocessing Server and Chatscript form one component of the CA system that is named "Conversational Agent Dialog System".

As mentioned before, the CA system needs to have a standalone frontend server and needs to retrieve data from the smaXtec REST API. The frontend will serve as a integration to already existing smaXtec web based user interfaces and as a standalone user interface for the administration panel. As seen in figure **??** the CA system is separated into two main components. These components are the "Conversational Agent Dialog System" component and the "Conversational Agent Frontend System" component. They interact with the smaXtec system via HTTP communication. Based on the integration requirements and decision the CA System does not have a direct internal communication channel with the smaXtec system, requests are send via the internet using HTTP requests form the CA backend and frontend components to the smaXtec REST API in order to retrieve user data.

5.6.1. Chatscript Component

This component runs an instance of the Chatscript framework that combines Natural Language Processing and Dialog Management in order to generate answers to user questions from the knowledge base. Since the server was implemented in the C programming language it is extremely fast, when compared to the python server. The main purpose of this server is to generate accurate answers to user questions in a acceptable time interval. This component receives user queries from the python preprocessing server, processes them and executes chatscript scripts to answer questions from the mentioned query.

5.6.2. PreProcessing Server

The objectives of this component are the most important. This server has to determine if there already exists an answer to a specific question, implements a clustering algorithm for questions, interacts with the API of the company and saves user interactions with the chatbot. It is also the backbone of the backend system since it is used for the administration panel.

The main components of the Flask Preprocessing server are Natural Language Toolkit, SciPy, Flask API and SqlAlchemy. Each of these components has a specific task that it fulfills. The Flask API component handles API requests from the client applications and calls the correct functions. Processing user text messages is the task of the Natural language toolkit. The SciPy component compliments the Natural language toolkit by implementing simple machine learning algorithms for clustering of user requests. The last component of the Flask preprocessing server is the SqlAlchemy python library, it is an object relationship management library that is used to communicate with the internal PSQL database that the system uses.

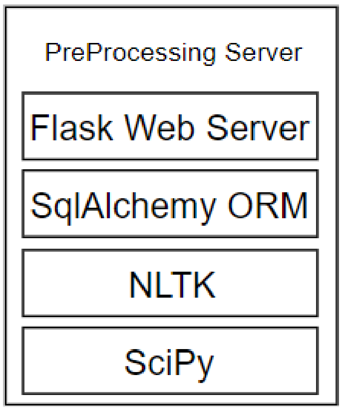

Figure 5.7.: PreProcessing Server Component

5.6.3. Database Structure

A standalone database is needed to preserve user conversations and question-answer pairs. Text fields of variable sizes are used to save the question answer pairs into the database. The best way to save question-answer pairs is to group them into topics for faster retrieval and better overview of questions asked by users. Since one answer can be an answer for multiple questions and multiple questions can have one answer, it is noticeable that the question-answer data structure is a many to many database connection. This means that one question can be a parent object to multiple answers, also a question topic can be parent object to questions. In this case they form a question cluster, with all related questions of a topic stored together. This is why database is formed out of three tables, a cluster table, cluster entry table and a link table for clusters and cluster entries.

The relationships between the entities are clearly visible and imply the usage of a relationship database management system[1] (Codd, 1990).The database structure diagram with the relationships between entities is visible in figure 5.8.

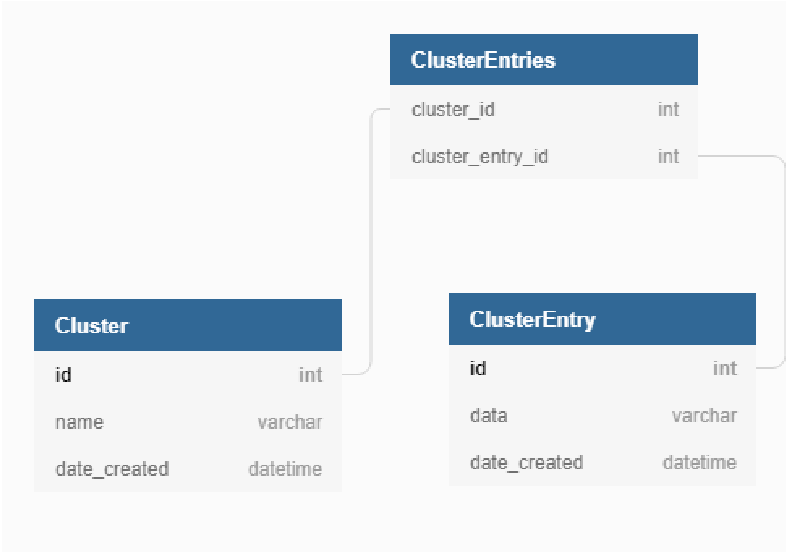

Figure 5.8.: Database Structure

5.6.4. Conversational Agent Frontend System

The frontend user interface consists of a chat interface for the customer and a administration panel interface for the customer employee. The purpose of the administration panel is to enable extendability of the chatbot system by allowing the user to answer unanswered questions or to add new question-answer pairs, thus expanding the knowledge base of the chatbot system. The requirements of the company described in chapter 2 require the whole frontend system to be implemented in AngularJS. The purpose of the chat interface is to enable communication between the user and the chatbot agent. While the administration panel is used to extend the capabilities and knowledge base of the chatbot agent. From the usability perspective, the chatbot interface should reduce the effort of the user to find the information that he is searching for. Instead of browsing the web page for answers to specific questions, he should have the ability to ask the chatbot agent directly.

5.6.5. Chatbot System and smaXtec System Architecture

In the previous sections individual components of the CA system architecture were discussed, this section will describe in details the whole architecture of the CA and

[1] https://www.techopedia.com/definition/24361/database-management-systems-dbms

how it integrates into the existing smaXtec system architecture. Figure 5.6 describes how the CA system integrates into the existing smaXtec system. Two main questions are considered here, how to integrate the CA System into the customer support area and information retrieval section of the smaXtec system.

Based on figure 5.6 and section 5.6 it is known that the goal of the chatbot backend system is to retrieve information from the smaXtec backend system and an internal database. The chatbot backend system uses HTTP requests to communicate with the smaXtec backend system. This communication enables the chatbot backend system to retrieve information from the smaXtec database. In order to uphold the non-function requirements of scalabiltiy, it is necessary that the CA system can be redundant. Redundancy[1] is the duplication of components of a system to increase the reliability of the system. This is why the chatbot backend system consists of multiple chatbot backed components. Each chatbot component communicates with the smaXtec backend system and a centralised chatbot database. Since the state of these components is preserved on the chatbot database, multiple instances of the chatbot system can be started at the same time. Each chatbot backend components consists of a chatscript server and a preprocessing server, which have been described in the previous chapters.

The chatbot frontend is a standalone component which communicates with the chatbot backend system. This component can run as a standalone component, meaning it can run on a server and be reachable via a custom URL endpoint, since it is a web application. It can also be embedded into an existing front end system. The goal of the chatbot frontend component is to provide a chat user interface, where the user can communicate via natural language with the chatbot system. This component together with the iframe is crucial to the integration of the chatbot system with already existing systems. An inline frame or iframe is used to embed another HTML document within the current HTML document. To integrate the chatbot system into the existing smaXtec frontend system or any other frontend system, it is necessary to create an iframe that opens the chatbot frontend system inside an existing web application. This integrability of the frontend system enables the CA system to implement the requirements to be used in both the information retrieval and customer support systems.

5.7. Summary

Technologies and concepts mentioned in the previous chapters have been used to formulate a prototype concept and architecture that meets the requirements stated in this chapter. This chapter described the process of creating the architectural concept of

[1] https://en.wikipedia.org/wiki/Redundancy_engineering

the CA based on the analysis of the current system and requirements. The main functional requirements for the prototype are that the chatbot system must respond when a conversation is initiated and must work on a web client(e.g. via mobile phone or desktop computer), the user should be able to communicate with the chatbot system in natural language. The natural language of choice is the English language. Maintenance of the knowledge base is an important aspect of the prototype. This implies that maintainers should be able to create questin answer pairs and answer unanswered questions.

A Conversation Agent based on NLTK and Chatscript was planned. This CA system consists of two main components: Conversational Agent Frontend System and Conversational Agent Dialog System. The Conversational Agent Dialog System used the Chatscript framework together with a Python Flask Processing Web Server. A cluster-based PSQL database structure was suggested to fulfill all requirements of the system and to make the system performant. To make the scalable and redundant, an architecture was purposed where the number of PreProcessing servers and Chatscript instances can be increased without affecting the system operations.

The Conversational Agent Frontend System component is the most important component for the integration of the system into the existing smaXtec system. This component allows the smaXtec frontend component and customer support components to communicate with the chatbot system.

The next chapter describes the implementation of the prototype planned above.

6. Development

This chapter goes into details about the technologies that were used in the prototype and explains the implementation process. Besides the features, the processes of the system will be explained together with the data structures used in the user request and communication between different components of the system.

Figure 5.6 describes the two main components of the system. The first and most important one is the Conversational Agent Dialog System that consists of Chatscript and a PreProcessing server and the Conversational Agent Frontend System component. The following sections explain in details the functions of these components.

6.1. Conversational Agent Dialog System

This section focuses on the implementation details and in-depth technological description of the Conversational Agent Dialog System and its components. Detailed processes and user interaction within the Conversation Agent System are also explained in this section.

6.1.1. Chatscript

Chatscript is a framework that combines Natural Language Processing and Dialog Management implemented with the main goal being the creation of Chat-bots. The framework was initially written in C++ and is now an open-source project that multiple contributors working on it. Companies use Chat script to create their Chat-bots because the past 4 Loebner prize winner Chatbots were written in Chatscript. Due to the nature that the framework was written in C++ it offers low response times and fast execution time. Chatscript is a framework that is based on program scripts. These scripts contain rules because the Chatscript engine is a rule-based engine. These scripts are written by individuals and the process of writing these scripts is called dialog flow scripting. Because these scripts are used to define the flow of the dialog. (Bruce Wilcox, 2013). Figure 6.1 shows a chatscript script with all rule types included for the topic food. A script that is 32 lines long can cover a longer conversation between a chatbot and an individual about food.

```
topic: ~food []

#! Hello
s: ( Hello ) Hello, do you need my help with something?

        a: ( ~yes)  How can I help you.
                    Do you need information about smaXtec?
            b: ( ~no ) Okay, how can I help you then.
            b: ( yes ) The company smaXtec is great.

        a: ( ~no) Well I can do a lot of things?
                b: ( What ) I can retrieve information.

#! What is smaXtec
u: ( what * smaXtec ) smaXtec is a company dedicated to animal care.

#! Where is smaXtec based?
?: ( location * smaxtec ) Smaxtec is based in Graz

#! do you like smaXtec or Tu Graz?
?: ( do you like _* or _* ) I do like '_0 but
I guess that means I also like '_1.

#! I like smaXtec.
s: ( ~like smaXtec )  Wow, that is nice...

#! How much do you work?
?: ( much * work ) I work always I never get tired
```

Listing 6.1: Chatscript Script Sample

A chatscript script always starts with a topic tag that serves as a container for communicational patterns. Every topic contains a set of rules, the rules are used to define the behavior of the chatbot. Rules start with "t:", "?:", "u:" or "s:". Reaction to statements is triggered with a "s:" rule. Questions reactions are described with "?:". In order for the chatbot to react to questions and statements "u:" is used. The symbol "t:" is a rule called a gambit, which enables the chatbot to react when he has no defined answers for questions in order to keep the conversation going..

The basic features that chatscipt provides show how strong of a framework it is for designing chatbots. Chatscript works on the following platforms, for personal computers Windows or Linux or Mac and for mobile devices, it runs on iOS or Android.

One Chatscript build-in function that was regularly used in the implementation of the prototype was "^jsonopen", which allows contacting an external service via HTTP. It was utilized in this prototype to define the communication channel between the Chatscript service and the smaXtec web application.

All files affiliated with a particular CA are stored in one folder. These files might include different topics files, the topic files always end with ".top". In the same directory as this folder, a ".txt" file is necessary, this ".txt" file lists all the paths to all the files of a specific CA. This text file is used when compiling the rules and topics of the CA. The folder structure for the smaXbot CA can be seen in figure 6.1.

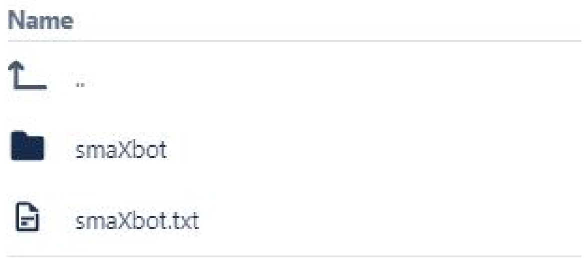

Figure 6.1.: smaXbot Chatscript Structure

The smaXbot folder from figure 6.1 contains all topic files related to the smaXtec system and can be extended automatically from the administration panel, but the chatscript component has to be restarted every time a new topic file is added for it to affect.

6.1.2. Flask and PreProcessing Server

Flask is small web framework written in python, because it does not require particular tools and libraries it is classified as a microframework. Flask does not have a database layer for object-relational mapping, tools like form validation, or any other function where pre-existing libraries provide these common functionalities. This implicates that Flask has no native support for the execution of the formerly mentioned common functionalities. These features and many other features can be added through extensions. This enables the developer to carefully pick features and extensions that are needed for a project, in contrast with large web frameworks, where these features do come out of the box with them. Flask can be installed with the usage of the python package manager (PIP), this is also the easiest way to install flask. In order to prevent cluttering and interference in the main running operating systems, flask is installed in virtual environments. Virtual environment is a copy of the Python interpreter in which you can install necessary python packages without interaction with the global python interpreter on your system. Virtual environments and the installation of Flask on separate virtual environments enables the applications to have access to the packages they need and not others.

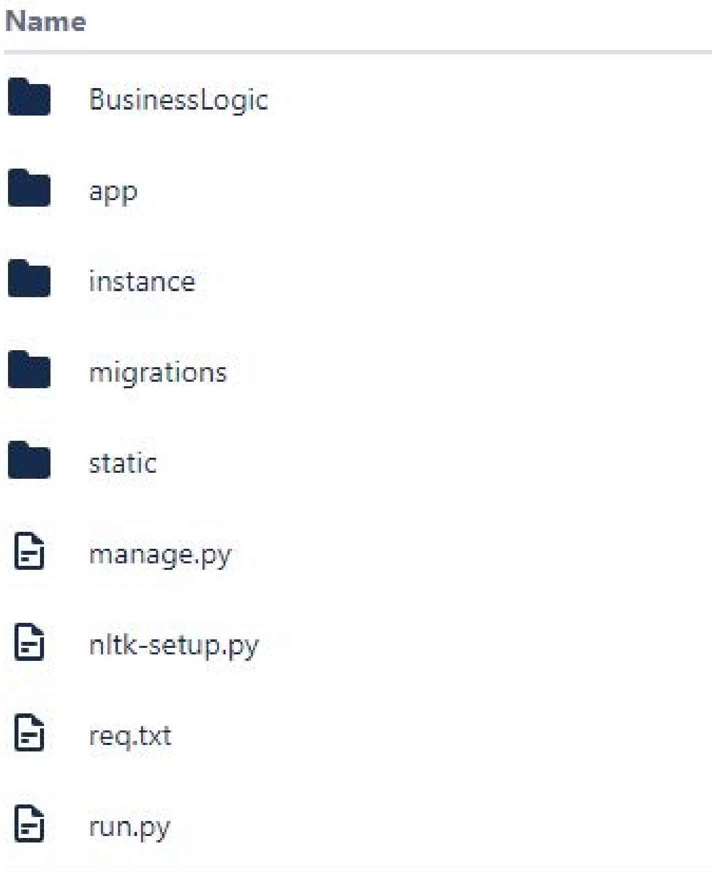

Figure 6.2.: PreProcessing Server Project Folder Structure

The PreProcessing Server has the structure as shown in figure 6.2. The main components of structure are the ”BusinessLogic” and ”app” compoenents. The ”BusinesLogic” folder contains all functions that are used for natural language processing. The ”app” folder contains the configuration files of the REST API and uses the business logic to process and to respond to user requests. The ”manage.py” and ”nltk-setup.py” scripts are used for code maintenance, the ”nltk-setup.py” downloads all necessary NLP compoenets that the ”BusinesLogic” uses and the ”manage.py” script is used to generate database specific code. Table 6.1 show all REST API calls that the Flask PreProcessing Server can handle.

Component	URL	Parameters	Description
NLP	/api/text	text	Extracts POS tags and sentiment from text
NLP	/api/textextra	text	Extracts named entities from string
NLP	/api/cosinesimilarity	text1 text2	Compares two strings to determine the similarity between them
CLUSTER	/api/clusters/add	name type	Creates a new cluster in the internal database
CLUSTER	/api/clusters/delete	id	Deletes cluster from database with given id
CLUSTER	/api/clusters/userstatement/add	clusterId message	Adds a user message to cluster with given id
CLUSTER	/api/clusters/clusterId/statement/statementId	clusterId statementId	Assigns a user statement to a cluster
CLUSTER	/api/clusters/types		Retrieves all cluster types
CLUSTER	/api/clusters/addtype	name	Creates a new cluster type with given name
CHATSCRIPT	/api/chatscript/	clusterId	Generates a Chatscript script from a cluster
CHAT	/api/chat/userstatement/add	user message	Generates a response to user message

Table 6.1.: PreProcessing Server REST API Endpoints

6.1.3. NLTK

In early 2001 Natural Language Toolkit or short NLTK was created by Steven Bird and Edward Loper in the Department of Computer and Information Science at the University of Pennsylvania. NLTK is a framework written in python and is contains multiple libraries and python programs that are used for natural language processing and tasks related to natural language processing. Besides these libraries and python programs, NLTK contains more than 50 corpora and lexical resources. NLTK functionalities can be seen in table 6.2

Language processing task	NLTK modules	Functionality
Accessing corpora	Corpus	Standardized interfaces to corpora and lexicons
String processing	Tokenize, stem	Tokenizers, sentence tokenizers, stemmers
Collocation discovery	Collocations	T-test, chi-squared, point-wise mutual information
Part-of-speech tagging	Tag	N-gram, backoff, Brill, HMM, TnT
Machine learning	Classify, cluster, tbl	decision tree, maximum entropy, naive Bayes, EM, k-means
Chunking	Chunk	Regular expression, n-gram, named-entity
Parsing	Parse, ccg	Chart, feature-based, unification, probabilistic, dependency
Semantic interpretation	Sem, inference	Lambda calculus, first-order logic, model checking
Evaluation metrics	Metrics	Precision, recall, agreement coefficients
Probability and estimation	Probability	Frequency distributions, smoothed probability distributions
Applications	App, chat	Graphical concordancer, parsers, WordNet browser, chatbots

Table 6.2.: NLTK Functionalities (Loper & Bird, 2002)

6.1.4. Database

As mentioned in chapter 5 the database is Postgres SQL based database. The database stores data about user queries, topics and responses to user queries. For the PreProcessing server to communicate with the database it was necessary to map the database tables onto Python objects to be used by the SQLAlchemy library to retrieve data from the database or to save data to the database. The database is a separate PSQL instance, this enables multiple PreProcessing instances to connect to the database and also makes the system extendible because other systems can connect to the same database to modify the database tables.

6.2. Process of Response Generation

The system is capable to a lead complex conversation with the user. The process of responce generation can be observed in figure 6.3.

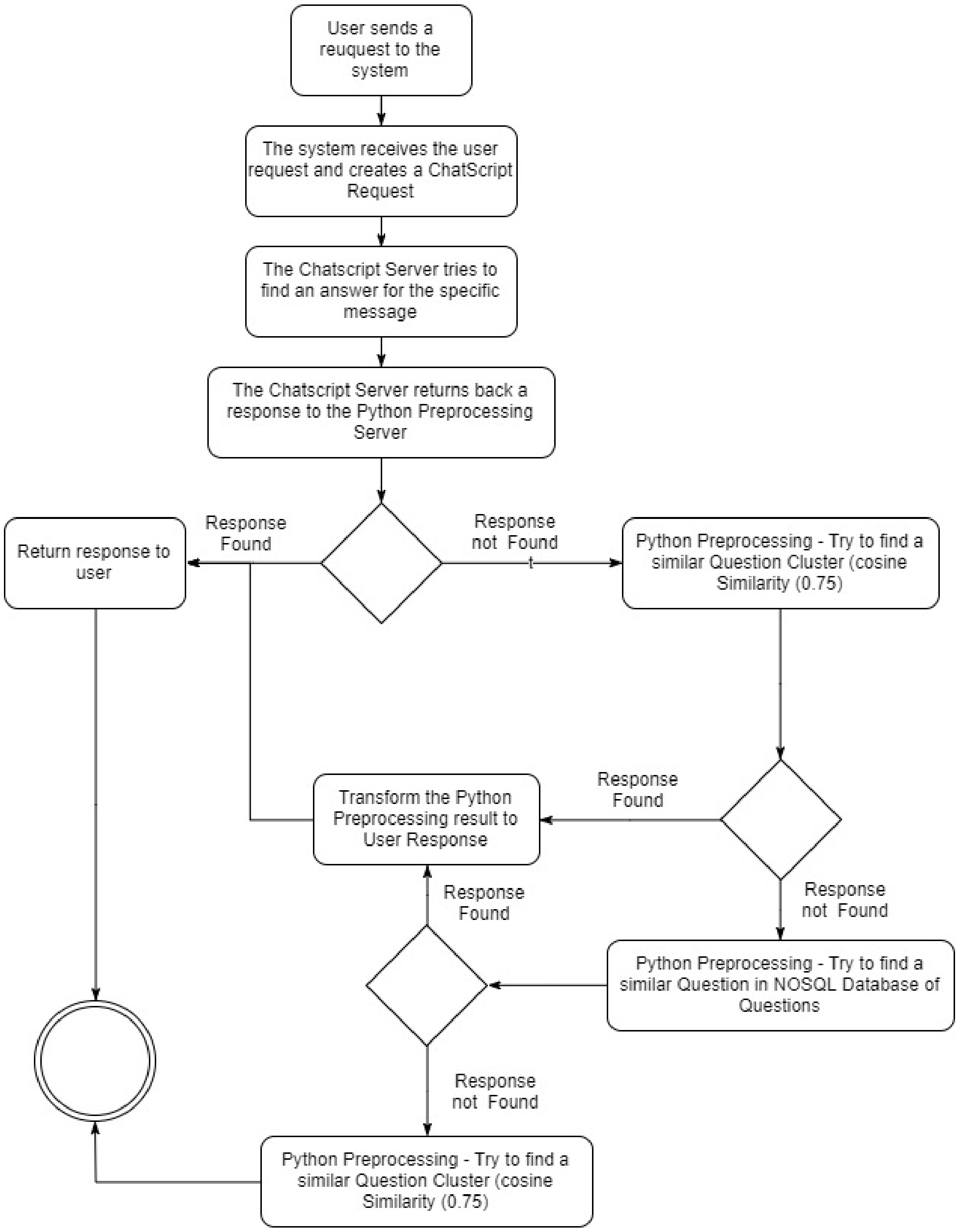

Figure 6.3.: The Process of Responses Generation

The process starts with a client sending a message to the python flask server. The python server receives the chat sentence, then forwards the request to the Chat script server with the same sentence. Chat script server will receive the full sentence, search for the response in the topic files and if the response is found it will return it to the python preprocessing server. If the response cannot be generated by the chatscript server, it returns "I do not know what to say" as the response. The python preprocessing server tries to find similar sentences in the existing question clusters with the usage of cosine similarity function (the threshold is 0.75). In the case that multiple sentence are found for the query, they are presented to the user as a multiple choice, where the user can choose which of those sentences is the correct one. In the case that the question is not found further processing is executed to find a similar question and provide the correct answer. The process of response generation relies on the effective cooperation of the python preprocessing server and the Chatscript server.

The Process of Responses Generation can be summarised into:

- Receiving the request
- Extracting the message, user and intent part of the message and NLP Analysis of the message
- Sending the results of the Chatscript analysis to the Python Preprocessing Server and Python Preprocessing Server response analysis
- Returning the answer

6.2.1. Receiving the Request

The client application sends a request in the form as shown in figure 6.2. The user part is used to identify the user sending the request and the message part of the request contains the chat sentence that the user sent. When the user writes a sentence in the chat client application it is sent to the Python Preprocessing Server.

```
{
"request": {
        "user": "user@smaxtec.com",
        "message": "hello"
    }
}
```

Listing 6.2: User Request

6.2.2. Extracting the Message, User and Intent Part of the Message and NLP Analysis of the Message

When the PreProcessing Server receives the request from the user it then parses it to extract the message and the user name. This data is forwarded to the Chatscript server for further processing. The Chatscript engine does the natural language processing of the message and saves user data to the file system. With the power of key-word search the engine tries to find the suitable topic and inside that topic the correct answer for the user request. In the case that the system is not able to find an answer it replies with "I do not know what to say.". Which is a system constant in the Processing Environment and signalizes that the preprocessing server has to do natural language processing and retrieval of information based on keywords.

6.2.3. Sending the Results of the Chatscript Analysis to the PreProcessing Server and PreProcessing Server Response Analysis

After the Chatscript server is done with the processing of the user request it sends the data back to the Preprocessing Server.

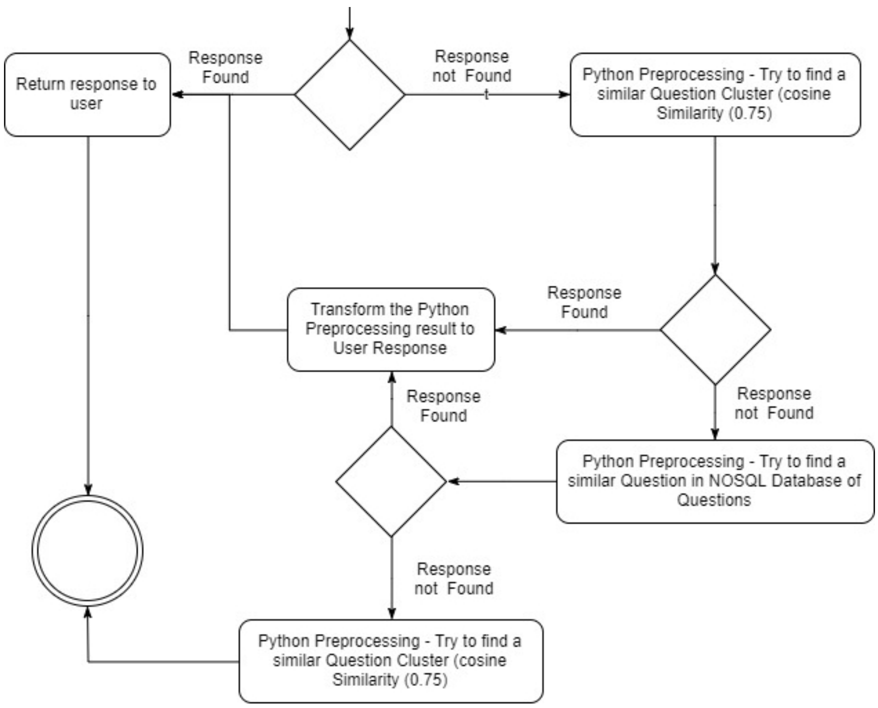

Figure 6.4.: PreProcessing Process

In the case that the result of the Chatscript processing is "I do not know what to say" the Preprocessing Server takes over and tries to generate a appropriate response. The start of the process is the search for similar phrases in the question clusters, with the usage of the cosine similarity and the similarity threshold of 0.75. If there are no similar phrases, the next step is the search for similar phrases in the NOSQL Database, which contains all messages that all users have sent to the system. Only in the case when there is a full overlap of the phrases the answer is phrase returned back as a result. When the first two steps fail the last step is the search for similar phrases in the question clusters with the usage of the cosine similarity but with a weakened threshold.

6.2.4. Returning the Answer

At the end of the process there is the step of the creating of the user response (answer). In the case that the Chatscript yields a valid response, the user receives the response directly from the Preprocessing Server, otherwise the results of the analysis have to be converted to a valid user response. Since the result of the similarity analysis can yield multiple results they have to be converted in a multiple choice object for the front end client.

In figure 6.11 this whole process is visible. The user tries to ask the chatbot a questions, which the chatbot is not able to answer. Instead of stopping the conversation it tries to initiate the user to reformulate the questions in order to find the right information for the user. After the user reformulates the question, the system tries to understand if the initial question that the user asked is also a formulation of the found answer.

6.3. Conversational Agent Frontend System

This section focuses on the implementation details and in-depth technological description of the Conversational Agent Frontend System and its components. The main components of the Conversational Agent Frontend are the Chat Interface and the Chat Agent Administration Panel.

6.3.1. AngularJS

AngularJS [1] is an open-source framework used for frontend development. It is based on JavaScript, the programming language used for the creating of web applications and web pages. As well as JavaScript, AngularJs is also used to create web pages and applications. The main feature of AngularJs is two-way data binding, which enables the

[1] https://angularjs.org/

developer to write simple code that solves tasks, that would have needed a plethora of code previously. Besides the two-way data binding, AngularJs used dependency injection to eliminate code duplication and to simplify the codebase. It is used to build high performing large web applications and also smaller projects because it is quite easy to learn AngularJs. Application made with AngularJs are easyto mantain.(Freeman, 2014).

Angular Concepts:

- **The Template** - The template is an HTML snippet that is used to describe what should be displayed to the user
- **The Directive** - The directive is used to create a small package that contains an HTML template and an operational Javascript code focused on executing one simple task.
- **The Scope** - The scope is the context of the application, it determines which controllers can be used and which templates are used for the application.
- **The Module** - The Module can be considered the collection of all scopes, directives, controllers, and scopes.
- **Data Binding** - Data Binding is used to bind specific variables to specific places in the HTML template.

AngularJS is a great framework that is used to create efficient single-page applications. It enables the developer to create a web application or web page using JavaScript, HTML, CSS while following the Model View Controller principle. AngularJs Applications are compatible with multiple browser versions. Since AngularJs is open-source it is completely free to be used and is used by a large number of developers around the world.

6.3.2. Bootstrap

Bootstrap is a free and open-source CSS framework aimed at responsive web development. It was created at Twitter headquarter, by Mark Otto and Jacob Thornton. The purpose of Bootstrap was to unify all tools that were used by Twitter developers. Bootstrap set the standard as the most used CSS framework aimed at responsive web development, eliminating inconsistencies and a high cost for maintenance (Walton, 2016).

6.3.3. Chat Agent Administration Panel

For administrating the data used by the CA, a simple administration panel has been created in AngularJS. The administration panel sends HTTP requests to the PreProcessing Server to execute different actions.The main operations that the administration panel fulfills are: Topic creation, assigning questions to topics, assigning answers to questions, removing questions from topics, modifying questions. The combination of these actions allows the administrators to improve the knowledge base of the chatbot administration system in an intuitive and uncomplicated way.

6.3.4. Usage of the Chat Agent Administration Panel

Figure 6.5 shows the overview page of the administration panel where the administrator can see different topics in which the CA can converse. This view allows the administrator to glance over the capabilities of the chatbot and determine which are he/she wants to improve. When the user opens the overview page an HTTP GET request is sent to the PreProcessing REST API via AJAX to retrieve all topics with question-answer pairs.

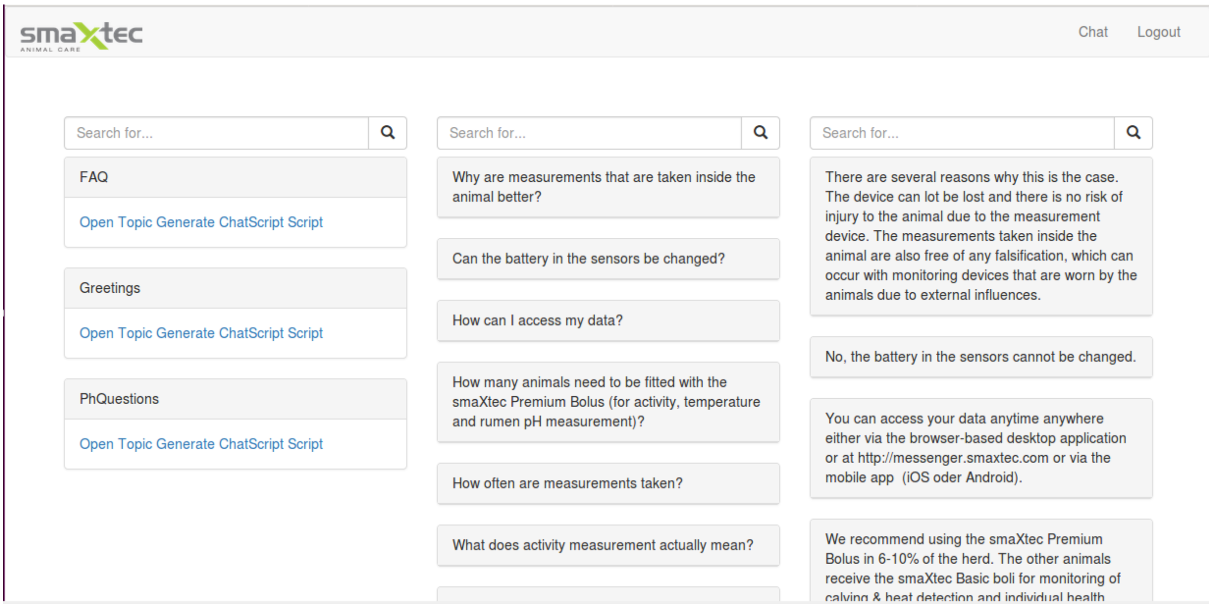

Figure 6.5.: Administration Panel Overview

When the user selects a topic by clicking on the topic name in the most left column, all questions related to the topic and the topic section are highlighted in blue. By clicking on the open topic link, the user is redirected to a topic overview page. The search bars above the columns enable the user to search for keywords in topics, questions, and answers.

The topic overview page can be seen in figure 6.7, this page contains all questions related to a topic. It also provides the administrator with the functionality to assign questions to the current topic and to edit and delete already assigned questions.

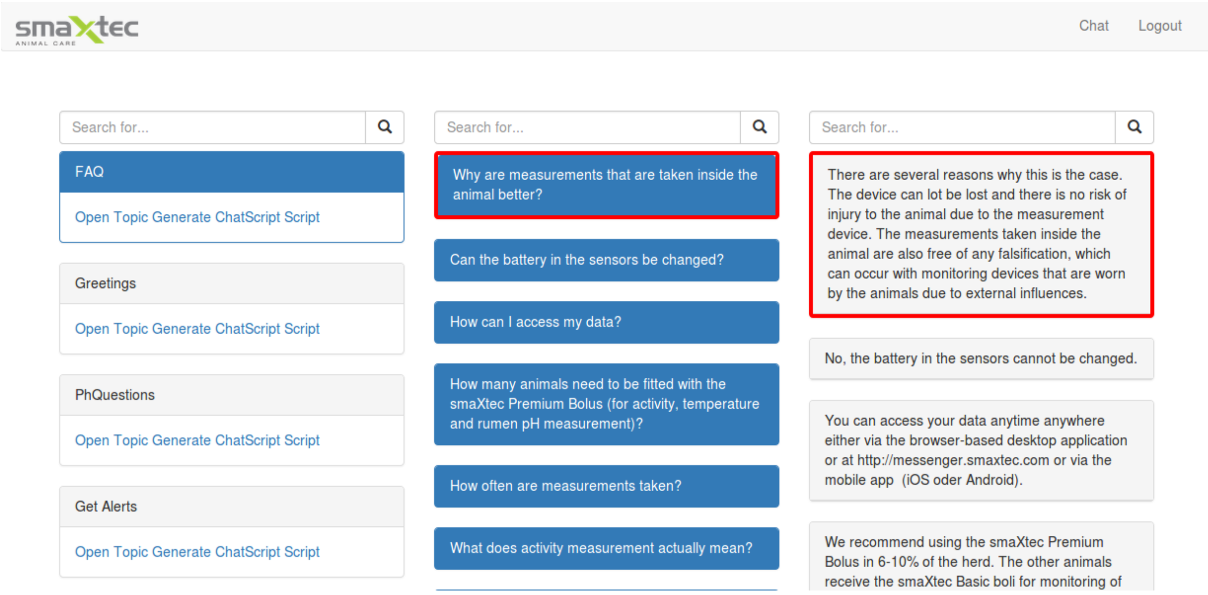

Figure 6.6.: Administration Panel Overview with Selection

By clicking the "Edit" button on the topic overview page (shown in figure 6.7) the user is prompted with the question edit popup as seen in figure 6.9.

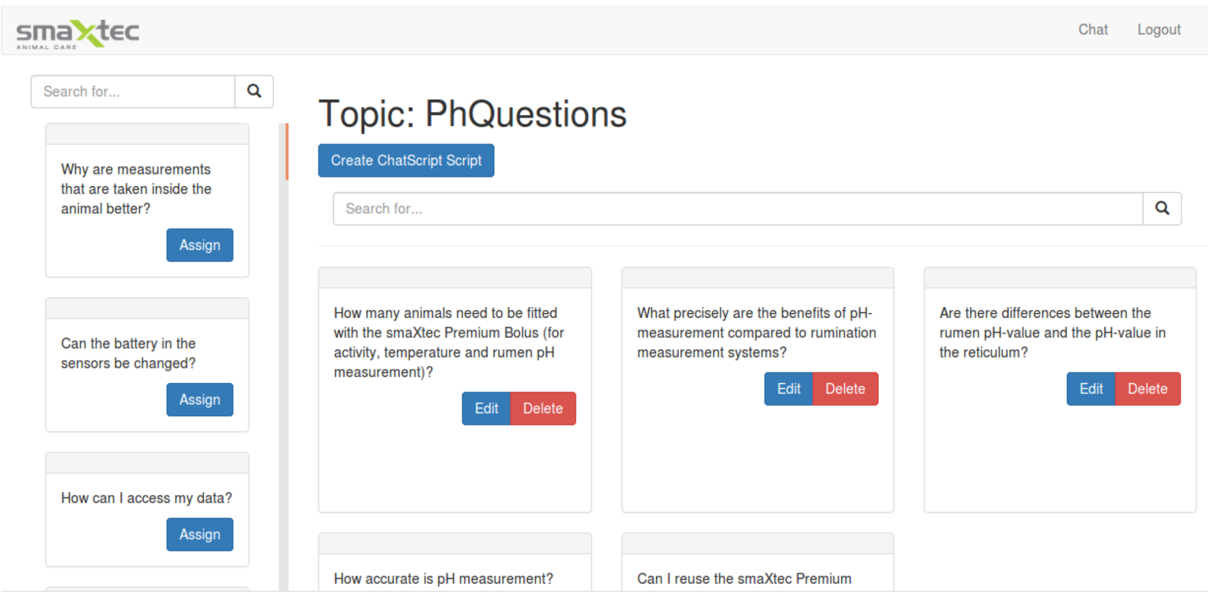

Figure 6.7.: Administration Panel Topic Overview

This window from figure 6.9 is used for adding question variations and answer variations. Since one question can be formulated in multiple ways, it is possible to assign multiple variations of questions to the original question and also to provide different answer variations to similar questions.

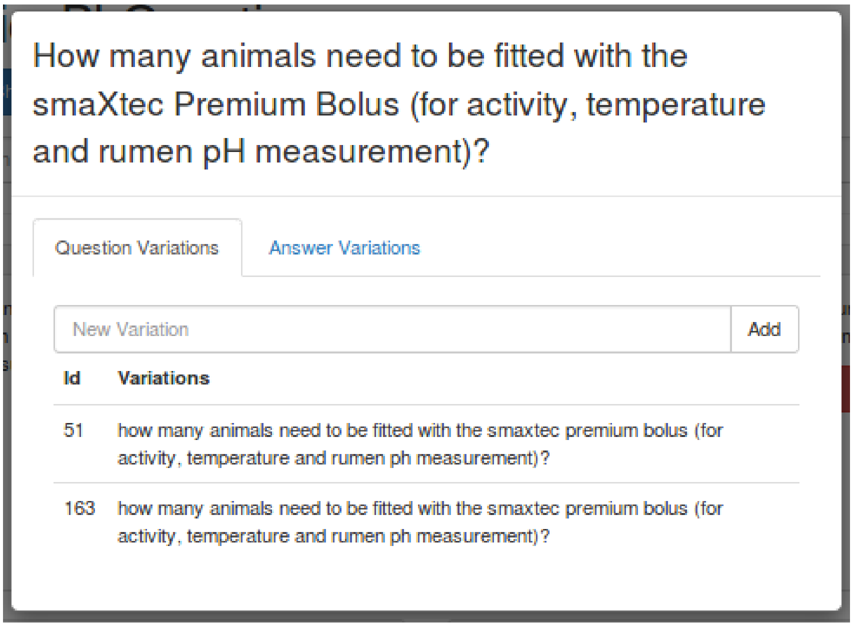

Figure 6.8.: Administration Panel Question Variation Edit

6.3.5. Creation of New Topic Clusters and Question-Answer Pairing

The main use-case of an administrator user is the creation of topic and assignment of question-answer pairs to the new topic. This use case starts with the opening of the administration panel and then clicking on the "New Topic" button. This opens the new topic creation page, the user is prompted to enter the name and description of the topic. The name and description are used to generate keywords for the new topic. When the user clicks "Create", he is redirected to the "Topic Edit" page, where he can see question answer pairs and where he can create new question answer pairs as seen in figure 6.7. If the administrator wants to create a new question-answer pair he needs to open the "New Question-Answer" window where he needs to input question variations and answer variations. At least one variation needs to be present for the question-answer pair to be saved. After the creation of the question-answer pair, the user needs to assign the question-answer pair to the new topic. The process can be seen in figure 6.9.

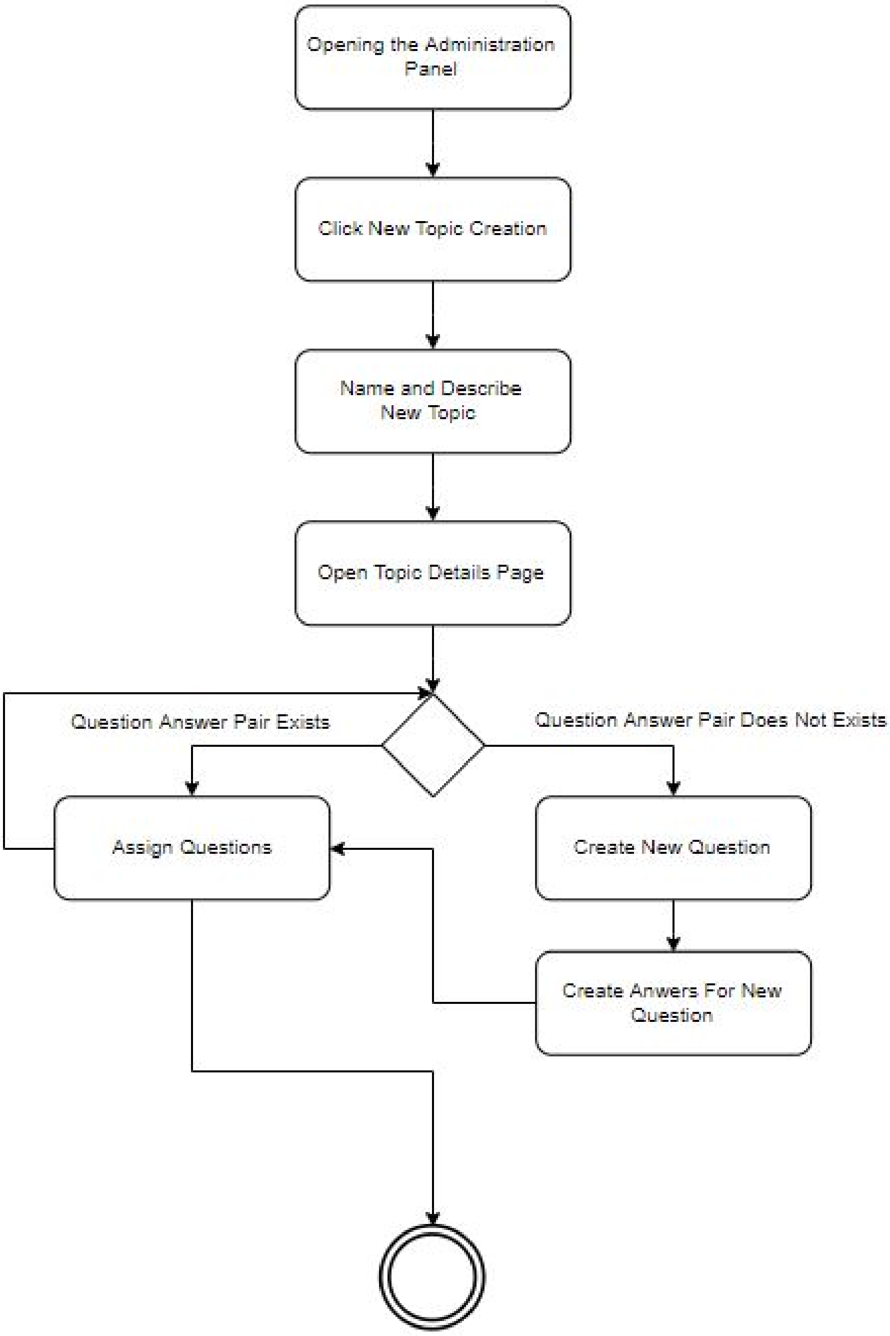

Figure 6.9.: Creation of New Topic Clusters and Question-Answer Pairing

6.3.6. Generation of ChatScript Scripts from Existing Topics

Generation of ChatScript scripts from existing topics is an importnat use-case. ChatScript Scripts are used to speed up the Natural Language Processing Workflow. To generate a new ChatScript script, the administrator has to open the Administration Panel on the ″Topic Details″ page where he can click on the ″ChatScript Script Generate″ button, which opens a new page with the ChatScript Script presented as a textual document. The user can download the script as a ″.top″ file. To update the ChatScript component with a new script he needs to copy the file to the ″Topic″ folder and restart the ChatScript Component. When the user has updated the ChatScript component, the PreProcessing server will no longer execute Natural Language Processing Scripts thus speeding up the Chat Agent answer generation process. The process can be seen in figure 6.10.

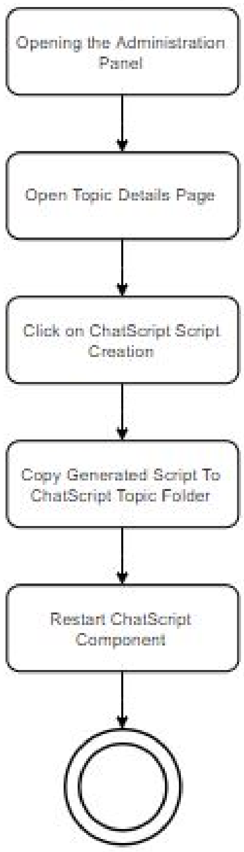

Figure 6.10.: Generation of ChatScript Scripts from Existing Topics

6.3.7. Chat Agent Chat Frontend

The Chat Agent Chat Frontend was implemented to simulate the looks of a chat application, where the user's main functionality is to create and send messages to other entities. The technologies used for the Chat Agent Chat Frontend were HTML, CSS, and AngularJS. These languages were interpreted by a web-browser and shown to the user either on a mobile or a desktop device.

Figure 6.11.: Chat Agent Chat Frontend

The Chat Agent Chat Frontend and the Chat Agent Administration Panel were implemented in the same frontend system in order to keep the project compact. Depending on where the Chat Agent Chat Frontend was integrated into, different options were available to the user.

In the case when the Chat Agent Chat Frontend has been setup as a stand-alone system, the administration panel would be visible to the user. In the case when the Chat Agent Chat Frontend was set up as a chat client, only the chat interface would be visible to the user.

6.3.8. Main Use Cases of the Chat Agent Frontend

Main Use-Cases as seen in figure 6.12 of the Chat Agent Frontend are related to the chat functionality of the Frontend. The user can ask a question in natural language. A

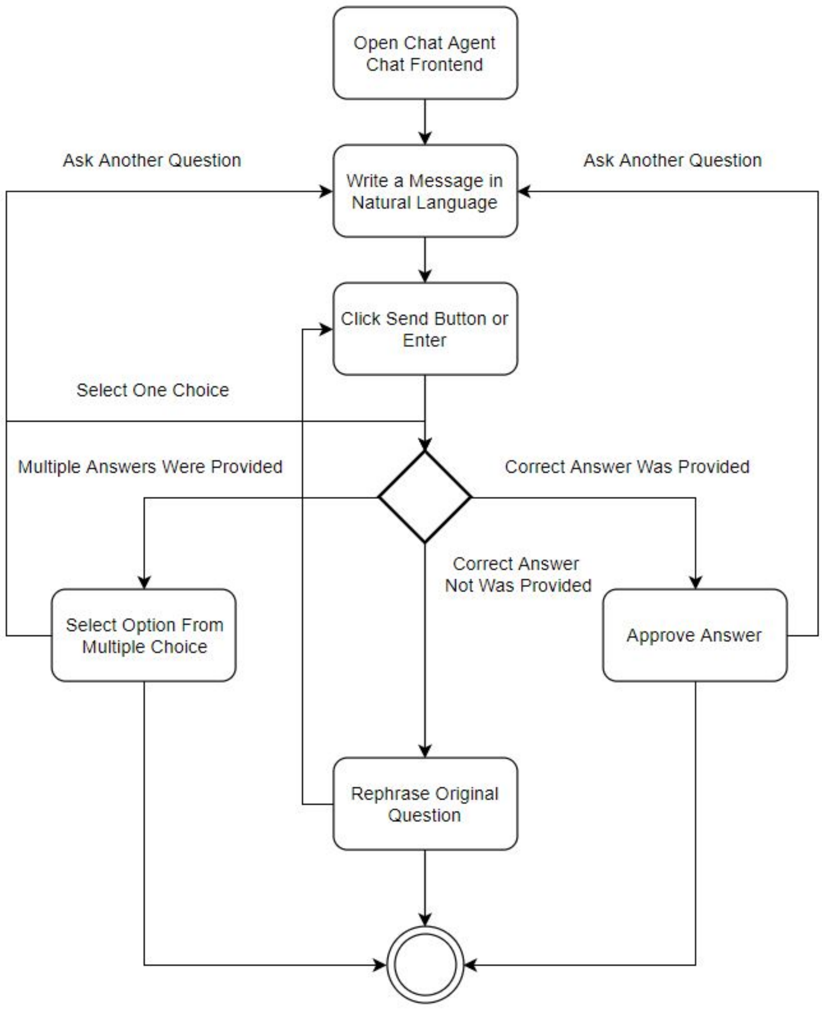

Figure 6.12.: Main Use Cases of the Chat Agent Frontend

sample question is "Can I get my Data", this is then sent to the PreProcessing server via AJAX and the user can get three different responses. One of these responses is a multiple-choice response, where the user can again select one of the multiple answers that were similar to his original question. The second option is that there is no answer to the asked question, which allows the user to rephrase the question or ask another question. The last option is the answer to the asked question, which can be approved by the user or the user might ask another question. These three options cover most of the interaction the user can have with the Chat Agent Frontend System to retrieve information.

6.3.9. Usage of the Chat Agent Chat Frontend

A detailed description of the interactions between the different components of the CA system in case of an entered sentence by the user is displayed in the figure 6.13. When the user enters a message and clicks on the "Send" button in the user interface, the AngularJs components retrieves that message and generates a HTTP request to the PreProcessing Server in order to retrieve a response from the system. The process of the reply generation is more explained in section 6.2.3.

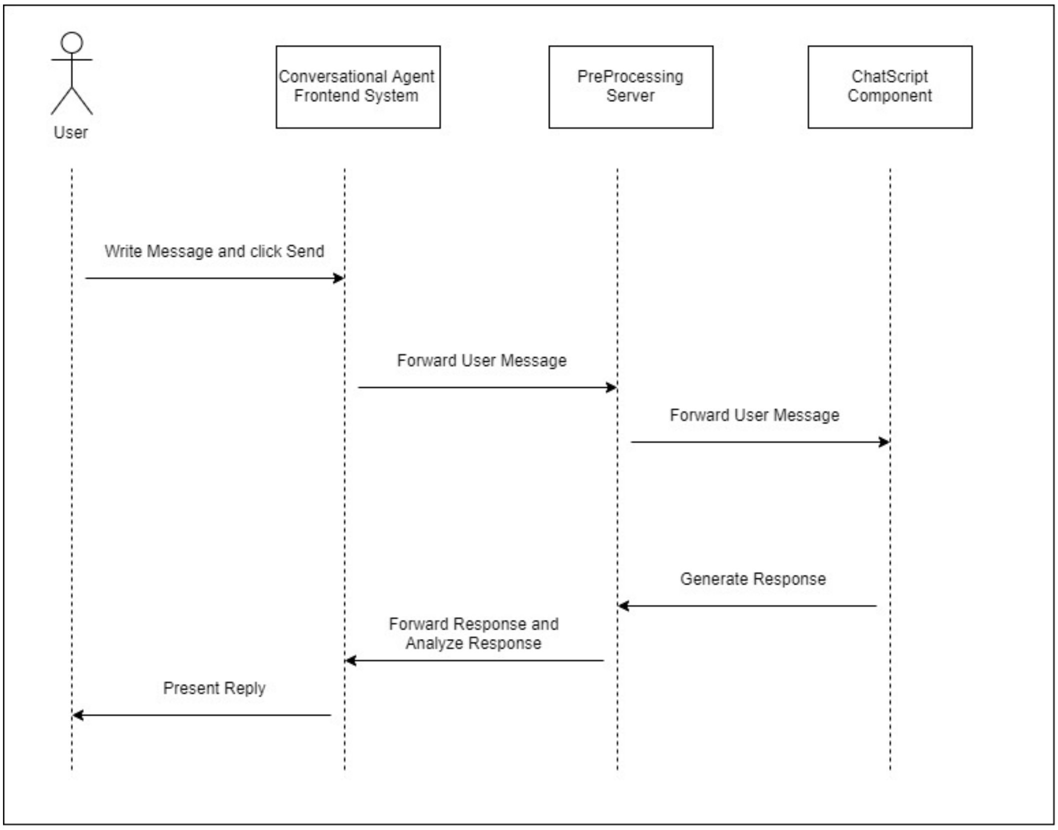

Figure 6.13.: Sequence Diagram Showing Reply Generation

6.4. Summary

Components of the CA system have been explained in detail together with implementation information. The most important processes of response generation and topic creation of the CA system have been explained in detail. Depending on the amount of information in the knowledge base in the Chatscript and python preprocessing system two different methods of response generation exist. In the case when the Chatscript system cannot generate a response for the user input, the python preprocessing server takes the role of natural language processing components and tries to generate a response. In the case that the Chatscript or python preprocessing system generates multiple responses, the user is given a choice to select the correct one.

To administer the knowledge base, an administration panel was created with the capabilities to create new topic clusters and assign new or existing questions to the subsequent clusters. These questions could contain different variations and would be linked to corresponding answers. The administration panel saves the changes only to the internal database, while the Chatscript knowledge base has to be updated manually.

The following chapter explains the study that was planned for testing the acceptance of the implemented prototype.

7. Evaluation

This chapter covers the evaluation of the previously mentioned Conversation Agent System. The selected evaluation tools are explained and the evaluation audience is described in detail. A thorough explanation of the evaluation process is given and the reasoning for the selection of specific evaluation tools is covered. The performance and usability of the previously mentioned system are discussed in this chapter.

7.1. Study Design

To evaluate the Conversation Agent System it was necessary to conduct a user study that measures the interaction of different individuals with the Conversation Agent System and compares it with the interactions with the existing smaXtec system. The user study was executed at the smaXtec company headquarters where several company employees and students from the Graz University of Technology used the prototype. Measuring user interaction with two different systems and enabling quantification of those interactions required the user study not to be exploratory. Specific tasks were assigned to the users that were performed with the Conversation Agent System and with the existing smaXtec system. Additionally, the goal was to evaluate how well do the users handle the new system, which concepts do they preferred and which concepts they dislike. The tasks are formulated in a way that the users must execute three different task types. The first task type is to find information related to the smaXtec and the smaXtec bolus which cannot be found in the frequently asked questions, this task type is called information search. The second task type is to find answers to frequently asked questions, this task type is called frequently asked question tasks. The last task is to retrieve specific information from the smaXtec system, this task type is called information retrieval tasks. The average execution of the task was limited to 2 minutes.

The study group was separated into two groups, one group would start executing tasks with the use of the existing smaXtec system, while the second group would start executing tasks with the Conversation Agent System. This was done to avoid bias towards one system. Because users would be more familiar with the task after they executed them once. Which would result in improved usage of the second system due to that familiarity.

The evaluation consisted of the following parts: user demographics and technical experience questionnaire, the execution of predefined tasks, rating of difficulty and information provided by the system after the execution of a task, questions for the System Usability Scale (SUS) (Brooke, 1996), questions for the Computer Emotion Scale (CES) (Kay & Loverock, 2008) and a feedback questionnaire. Logs for the conversations were kept to derive information about problems and usage behavior.

The goal of the user study is to provide insights in the following research questions:

1. Can a dialog system be used as an information retrieval system and a replacement for frequently asked questions pages?
2. Would the users utilize a dialog system to retrieve data and acquire answers to specific questions rather than an existing web based system?
3. Does the dialog system increase the speed of question-answering in comparison with the existing system?
4. Do the users feel comfortable with the usage of the dialog system?
5. Which emotions do they experience when using the dialog system?

7.2. Setting and Instrument

For the study, fourteen people were contacted consisting of 8 company employees and 6 students from the Graz University of Technology. The participants had to execute six specific tasks. The tasks can be found in table 7.1. Each participant had to do the evaluation individually and their behavior was observed to collect behavioral data about the participant. The participant had to talk while executing tasks to determine the motivation of their behavior.

Each evaluation started with a detailed introduction about the two systems, the study and what the participants should do related to the tasks. After this the participants were instructed to fill out a demographic and technical skill questionnaire, these questionnaire contain questions related to user age, education level and experience with chatbots and dialog systems. After successfully completing this questionnaire, the participant needed to execute the tasks mentioned in table 7.1, after the execution of all tasks the users had to rate the difficulty of each task on a scale from 1 to 5, where 1 is easy and 5 is difficult. This is used to rate the task difficulty. After the task difficulty rating the user had to rate how satisfied he/she was with the answer. This was also rated on a scale from 1 to 5, where 1 is satisfied and 5 is not satisfied. The purpose of this is to determine which system provides more satisfaction to the participant.

To measure the satisfaction of the participants and to evaluate how well the prototype was accepted by them, the Computer Emotion Scale (CES) and the System Usability Scale, were used in combination with additional feedback questions. These scales and the additional questionnaire were filled out after the execution of the specific tasks.

Task Number	Description	Task Type
1	What is smaXtec?	Information Search Tasks
2	What is a smaXtec bolus?	Information Search Tasks
3	How long does it take until the first data is available?	Frequently Asked Question Tasks
4	Does the smaXtec base station need an internet Link?	Frequently Asked Question Tasks
5	Get information about a cow with the name "Maria"	Information Retrieval Tasks
6	Get all alerts of today	Information Retrieval Tasks

Table 7.1.: Task For Participants

7.2.1. System Usability Scale

The System Usability Scale (SUS) is a scale formed of ten items on a Likert[1] scale. Every item contains a statement about the system and can be rated with a score between 0 and 4. Where 0 is used to express strong disagreement and 4 is used to express strong agreement. The System Usability Scale proved to be a strong and trustworthy evaluation tool. For calculation the score, the rating of questions 1,3,5,7 and 9 is subtracted by one and the result is summed up, then this sum is multipled by 2.5. The rating of questions 2,4,6,8 and 10 is subtracted from 5 and summed up, then multipled by 2.5. These two sums are then added up together resulting in a value between 0 and 100 (Brooke, 1996).

The scoring system shown in figure 7.3 was proposed to interpret the outcome of the SUS (Bangor et al., 2009). It is used right after the participants have completed the execution of the tasks and before the Computer Emotion Scale (CES).

[1] https://en.wikipedia.org/wiki/Likert$_{s}cale$

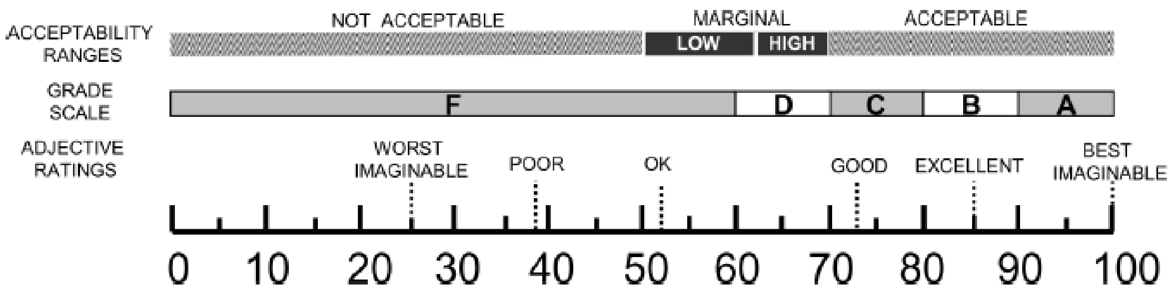

Figure 7.1.: System Usability Scale according to Bangor et al. (2009)

7.2.2. Computer Emotion Scale

The Computer Emotion Scale (CES) measures emotions that participants experience when using the Conversation Agent System. The focus was on four emotions during this user study: anxiety, anger, sadness, and happiness. The participants had to rate twelve feelings on a Likert scale with a score from 0 to 3. These feelings were grouped in the previously mentioned emotions as seen in table 7.2. The ratings were summed up for each emotion and normalized by dividing the sum with the number of feelings grouped to the selected emotion.

Emotion	Feeling
Anxiety	Anxious, Insecure, Helpless and Nervous
Anger	Irritable, Frustrated and Angry
Sadness	Disheartened and Dispirited
Happiness	Satisfied, Excited and Curious

Table 7.2.: Feeling to Emotion Grouping

The Computer Emotion Scale (CES) was selected because it produces comparable results and detects which emotions were the primary emotions during the user study, which can be a base for further studies. It is also be used to answer the research question "Do the users feel comfortable with the usage of the dialog system?" and "Which emotions do they experience when using the dialog system?".

7.2.3. Feedback Questionnaire

The last part of the user study is the feedback questionnaire as seen in A.1. This is used to determine what the users think about the system and to extract existing functionalities that work well and potential improvements to the system. The user is also asked to evaluate if they prefer working with the prototype system more than with the existing system.

7.3. Study Participants

The user survey group consisted of 15 participants, out of those 15 participants 10 were employees of smaXtec working in the Customer Support and IT area. The age of the participants varied from 20 to 34 years old. Female participants made 33.33% of the total amount of participant, 60% male and 6.66% prefered not to say. Participants with a Bachelor's degree made 60% of group, while 27% had finished Master's degrees, 7% were High School Graduates and 7% were individuals with a Doctoral degree. The group was highly technically skilled with 90 % of the working with computers. 90% of the participates had already had experiences with dialog systems, but the majority (50%) do not work on a daily basis with them. When asked the question "Which Dialog Systems did you use?" more than half of the participants answered with "Google". The satisfaction level of the participants in relation with dialog system was neutral, with 25% rating it with a 3 out of 5, 37.5% rating it with a 4 out of 5 and 37.5% rating it with a 3 out of 5.

Most of the users did not have positive comments when asked about experiences with dialog systems, with 80% of the users answering with "Nothing", Negative sides of dialog systems according to the user feedback were that the systems have a hard time understanding the commands, do not work as expected, provide wrong answers and that the answers are not understandable.

User Property	**Information**
Gender	Male 9 (60%), Female 5 (33.33%) and Prefer not to say 1 (6.66%)
The Higest Level Of Education	High School Graduation 1 (7%), Bachelor's Degree 9 (60%), Master's Degree 4 (27%) and Doctoral Degree 1 (7%)
Did you use dialog Systems earlier?	Yes 13 (87%) and No 2 (13%)
smaXtec Emloyee	10 (66.66%) Participants were smaXtec employees 5 (33.33%) were students at the Graz University of Technology

Table 7.3.: User Information

Each participant was randomly assigned the first and the second system to use for the execution of the previously mentioned tasks, 50% of the users started with the existing smaXtec system and 50% of the users started the evaluation with the Chat Agent System. It was noticed that the users could execute the tasks easier when the second system and executing the tasks for the second time.

Based on table 7.4, none of the participants works on a daily basis with dialog systems, four (26%) of the participants did not work at all with dialog systems. Most of the participants engage with dialog systems occasionally.

	1	2	3	4	5
How often do you work with dialog systems?	4 (26%)	8 (53.33%)	2(13.33%)	1(6.66%)	0(0%)
How oftern do you work with computers?	0(0%)	0(0%)	0(0%)	1(6.66%)	14(93.33%)

Table 7.4.: User Answers to User Specific Questions (1 - Not at all; 5 - On a daily Basis)

The satisfaction level of participants with dialog systems leans towards dissatisfaction, based on table 7.5, where 1 participant answered that he/she was very dissatisfied with the dialog systems that he/she already used. The majority of the participants were neither satisfied or dissatisfied with the usage of dialog systems.

	1	2	3	4	5
How satisfied were you with the dialog systems that you used?	1(6.66%)	3(20%)	6(40%)	3(20%)	0(0%)
How satisfied were you with the usability of the dialog systems?	0(0%)	5(33.33%)	4(26.66%)	4(26.66%)	0(0%)

Table 7.5.: User Answers to User Specific Questions (1 - Very dissatisfied; 5 - Very satisfied)

7.4. Findings and Discussion

In average the users needed 4 minutes and 24 seconds to complete the tasks when using the Chat Agent System, compared to 5 minutes and 48 seconds with the existing smaXtec system. Figure 7.6 describes the rating that the participants gave to the tasks when using the Chat Agent System.

Task	Very Difficult	Difficult	Moderate	Easy	Very Easy
What is smaXtec?	0 (0%)	0 (0%)	1 (6.66%)	2 (13.33%)	12 (80%)
What is a smaXtec bolus?	0 (0%)	3 (20%)	0 (0%)	3 (20%)	9 (60%)
How long does it take until the first data is available?	0 (0%)	0 (0%)	1 (6.66%)	9 (60%)	5 (33.3%)
Does the smaXtec base station need an internet Link?	0 (0%)	0 (0%)	5 (33.3%)	1 (6.66%)	9 (60%)
Get information about a cow with the name "Maria"	0 (0%)	1 (6.66%)	1 (6.66%)	4 (26.66%)	9 (60%)
Get all alerts of today	0 (0%)	0 (0%)	1 (6.66%)	5 (33.3%)	9 (60%)

Table 7.6.: Difficulty Rating of Tasks by Participants from Table 7.1 when executing the Tasks in the Chat Agent System

Figure 7.6 describes the rating that the participants gave to the tasks when using the smaXtec System.

Task	Very Difficult	Difficult	Moderate	Easy	Very Easy
What is smaXtec?	0 (0%)	3 (20%)	6 (40%)	3 (20%)	3 (20%)
What is a smaXtec bolus?	0 (0%)	4 (26.7%)	9(60%)	1 (6.7%)	1 (6.7%)
How long does it take until the first data is available?	1 (6.7%)	5 (33.3%)	3 (20%)	4 (26.7%)	2 (13.3%)
Does the smaXtec base station need an internet Link?	1 (6.7%)	4 (26.7%)	3 (20%)	6 (40%)	1 (6.7%)
Get information about a cow with the name "Maria"	0 (0%)	2 (13.3%)	3 (20%)	6 (40%)	4 (26.7%)
Get all alerts of today	0 (0%)	4 (26.7%)	8 (53.3%)	1 (6.7%)	2 (13.3%)

Table 7.7.: Difficulty Rating of Tasks by Participants from Table 7.1 when executing the Tasks in the smaXtec System

The first task was rated by 80 % of the participants as very easy when using the Chat Agent System and 20 % very easy when using the smaXtec System. The second task was rated by 60 % of the participants as very easy when using the Chat Agent System and 6.7 % very easy when using the smaXtec System. The third task was rated by 33.33 % of the participants as very easy when using the Chat Agent System and 13.3 % very easy when using the smaXtec System. The fourth task was rated by 60 % of the participants as very easy when using the Chat Agent System and 6.7 % very easy when using the smaXtec System. The fifth task was rated by 60 % of the participants as very easy when using the Chat Agent System and 26.7 % very easy when using the smaXtec System. The sixth task was rated by 60 % of the participants as very easy when using the Chat Agent System and 13.3 % very easy when using the smaXtec System as seen in tables 7.7 and 7.7. None of the tasks was rated as "Very Difficult" when using the Chat Agent System, while task number 3 and 4 was rated as "Very Difficult" by 6.7 % of the participants.

After the difficulty rating of the tasks, the participants also rated the answers that they found with the use of the Chat Agent System (7.8) and with the use of the smaXtec system (7.9).

Task	**Very Unclear**	**Unclear**	**Moderate**	**Clear**	**Very Clear**
What is smaXtec?	0 (0%)	0 (0%)	0 (0%)	4 (26.7%)	11 (73.3%)
What is a smaXtec bolus?	0 (0%)	0 (0%)	2 (13.3%)	5 (33.3%)	8 (53.3%)
How long does it take until the first data is available?	0 (0%)	0 (0%)	0 (0%)	2 (13.3%)	13 (86.7%)
Does the smaXtec base stationneed an internet Link?	0 (0%)	0 (0%)	2 (13.3%)	1 (6.7%)	12 (80%)
Get information about a cow with the name "Maria"	0 (0%)	0 (0%)	4 (26.7%)	4 (26.7%)	7 (46.7%)
Get all alerts of today	0 (0%)	0 (0%)	4 (26.7%)	1 (6.7%)	10 (66.7%)

Table 7.8.: Participant Rating of Information Clarity Of Answers Provided to the Tasks from Table 7.1 by the Chat Agent System

Comparing the two tables it is visible that the Chat Agent System provided clearer answers to the participates, while the smaXtec system provided moderate results at most. None of the answers of the Chat Agent System were unclear, while the smaXtec system produced unclear results in 26.7 % of the cases when executing the task "What is a smaXtec bolus?".

Task	Very Unclear	Unclear	Moderate	Clear	Very Clear
What is smaXtec?	0 (0%)	0 (0%)	3 (20%)	8 (53.3%)	4 (26.7%)
What is a smaXtec bolus?	0 (0%)	4 (26.7%)	5 (33.3%)	3 (20%)	3 (20%)
How long does it take until the first data is available?	0 (0%)	0 (0%)	3 (20%)	8 (53.3%)	4 (26.7%)
Does the smaXtec base station need an internet Link?	0 (0%)	0 (0%)	2 (13.3%)	9 (60%)	4 (26.7%)
Get information about a cow with the name "Maria"	0 (0%)	0 (0%)	3 (20%)	7 (46.7%)	5 (33.3%)
Get all alerts of today	0 (0%)	0 (0%)	4 (26.7%)	7 (46.7%)	4 (26.7%)

Table 7.9.: Participant Rating of Information Clarity Of Answers Provided to the Tasks from Table 7.1 by the smaXtec System

Most of the answers provided by the Chat Agent System were classified as very clear, with the third and fourth task having a clarity above 88.9%. The minimum percentage of "Very Clear" answers of the Chat Agent System is much higher than the maximum percentage of "Very Clear" answers of the smaXtec system according to participant reviews of tasks based on table 7.8 and 7.9.

7.4.1. System Usability Scale

Figure 7.3 describes the distribution of the answers for the SUS. It is visible that most of the users agree or strongly agree with questions 1, 3, 4, 5 and 7 while disagreeing or strongly disagreeing with the rest of the questions.

Due to the large number of neutral answers the average rating of the scale is 78.66 which is slightly above the limit of 70 set by Bangor et al. (2009) as the value that is the minimum for a usable system. The highest score was 97.5 and the lowest score was 50, which shows a variance of 47.5 points. The scores were calculated according to the explanation in section 7.2.1.

Figure 7.3 describes the distribution of points per participant for all fifteen participants and shows the average point as a reference. It is observable that the participant rating differs a lot per participant, with some participants scoring the system with as low as 50 and others scoring with 97.5.

	strongly disagree	disagree	neutral	agree	strongly agree
I think that I would like to use this system frequently	0 (0%)	0 (0%)	4 (26.7%)	7 (46.7%)	4 (26.7%)
I found the system unnecessarily complex.	6 (40%)	7 (46%)	2 (13%)	0 (0%)	0 (0%)
I thought the system was easy to use.	0 (0%)	0 (0%)	3 (20%)	2 (13%)	10 (66.7%)
I think that I would need the support of a technical person to be able to use this system.	10 (66.7%)	2 (13%)	3 (20%)	0 (0%)	0 (0%)
I found the various functions in this system were well integrated.	1 (6.7%)	0 (0%)	3 (20%)	3 (20%)	8 (53.3%)
I thought there was too much inconsistency in this system.	10 (66.7%)	3 (20%)	2 (13%)	0 (0%)	0 (0%)
I would imagine that most people would learn to use this system very quickly.	0 (0%)	0 (0%)	3 (20%)	9 (60%)	3 (20%)
I found the system very cumbersome to use.	3 (20%)	8 (53.3%)	3 (20%)	1 (6.7%)	0 (0%)
I felt very confident using the system.	0 (0%)	0 (0%)	4 (26.7%)	9 (60%)	2 (13%)
I needed to learn a lot of things before I could get going with this system.	3 (20%)	8 (53.3%)	3 (20%)	1 (6.7%)	0 (0%)

Table 7.10.: Number of times (percentage) an answer on the System Usability Scale has been selected

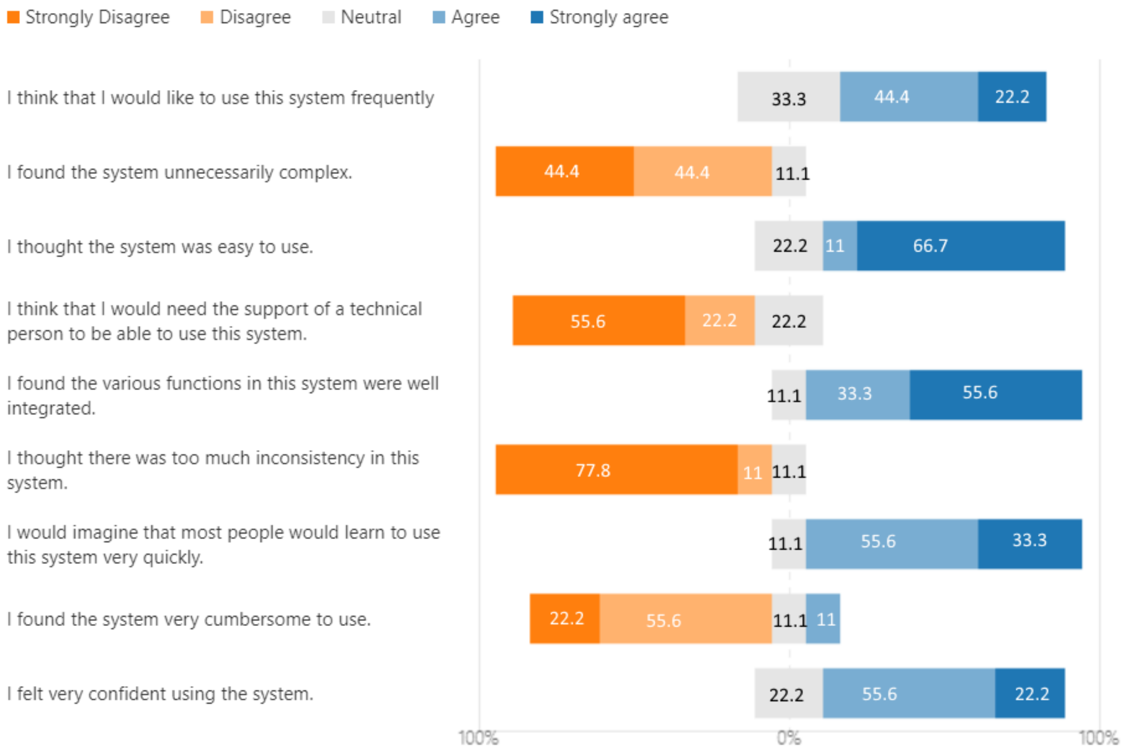

Figure 7.2.: Results of the System Usability Score

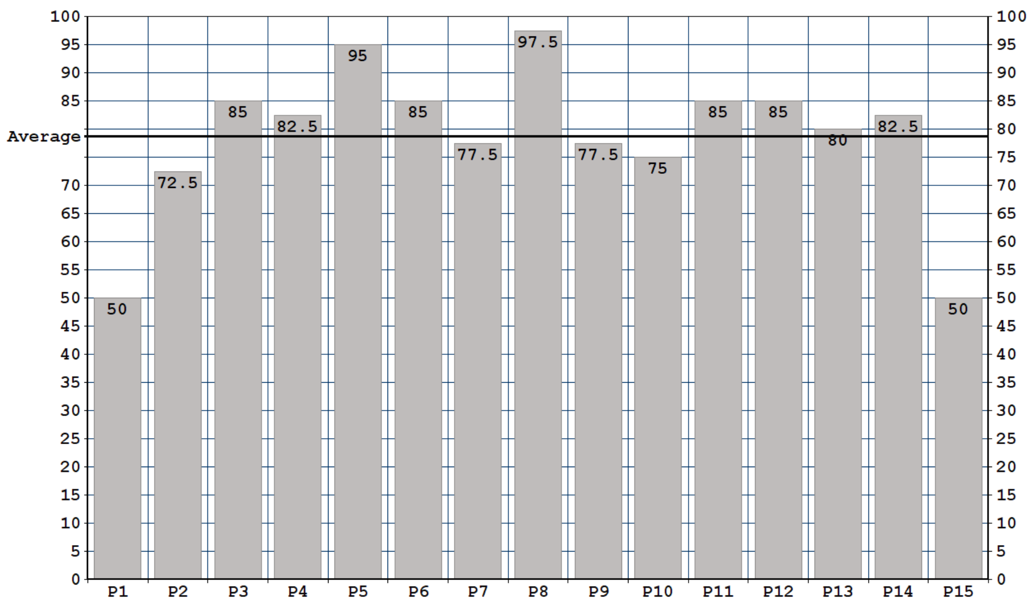

Figure 7.3.: Results of the System Usability Score per participant P1 – P15.

7.4.2. Computer Emotion Scale

The result of the CES is shown in table 7.11, the table contains a list of feelings a participant has experienced and shows the per cent of users that have experienced a specific emotion. The CES shows that the users were most of the time happy when using the Chat Agent System, while non of the time experiencing sadness, anxiety and anger. Based on table 7.11 the emotion anxiety has the lowest score because most of the users rated it with "non of the time", followed with Sadness and Anger. The best rated emotion was Happiness where the majority of the users answered with either "Some of the Time", "Most of the Time" or "All of the Time".

	None of the Time	Some of the Time	Most of the Time	All of the Time
satisfied	0 (0%)	0 (0%)	8 (53.3%)	7 (46.7%)
excited	1 (6.7 %)	11 (73.3%)	2 (13.3%)	1 (6.7%)
curious	0 (0%)	0 (0%)	12 (80%)	3 (20%)
Happiness	1 (2.22%)	11 (24.44%)	22 (48.88%)	11 (24.44%)
disheartened	12 (80%)	3 (20%)	0 (0%)	0 (0%)
dispirited	13 (86.7%)	2 (13.3%)	0 (0%)	0 (0%)
Sadness	25 (73.52%)	5 (18.75%)	0 (0%)	0 (0%)
anxious	15 (100%)	0 (0%)	0 (0%)	0 (0%)
insecure	12 (80%)	3 (20%)	0 (0%)	0 (0%)
helpless	7 (42.9%)	6 (42.9%)	1 (7.1%)	1 (7.1%)
nervous	15 (100%)	0 (0%)	0 (0%)	0 (0%)
Anxiety	49 (81.63%)	9 (15%)	1 (1.66%)	1 (1.66%)
irritable	10 (66.7%)	5 (33.3%)	0 (0%)	0 (0%)
frustrated	11 (73.3%)	4 (26.7%)	0 (0%)	0 (0%)
angry	11 (73.3%)	4 (26.7%)	0 (0%)	0 (0%)
Anger	32 (71.11%)	13 (28.88%)	0 (0%)	0 (0%)

Table 7.11.: Number of times (percentage) an answers has been selected on the Computer Emotion Scale

From table 7.11 it is visible that the participants felt happiness in 48.88 % most of the time, which is more than any other emotion. It is visible that happiness was the dominant emotion (41.66% most of the time and 29.16% all of the time) that the users felt during the usage of the Chat Agent System. Which in conclusion describes that the participants adjusted well to the Chat Agent System. While using the Chat Agent System the users experienced the feelings of curiosity and excitement based on table 7.11.

Figure 7.4 shows the distribution of user answers grouped by emotions. Anger (71.11%), anxiety(81.67%) and sadness (73.53%) were not felt by the majority of the participants, for the emotion happiness the answer that it was felt none of the time was only selected in 2.22% of the time and the answer that it was felt in most of the time during the usage of the Chat Agent system was selected in 48.89 % of the time.

Emotion	Average Rating
Happiness	1.95
Sadness	0.5
Anxiety	0.23
Anger	0.28

Table 7.12.: Average rating of all users per emotion

According to table 7.12, the emotion anxiety had the lowest average rating of 0.23, followed by anger with 0.28, then sadness with 0.5 and lastly happiness with 1.95. From this table it is visible that the majority of the participants were satisfied and happy with the use of the Chat Agent Prototype, while experiencing anger anxiety and sadness in much lower levels than happiness.

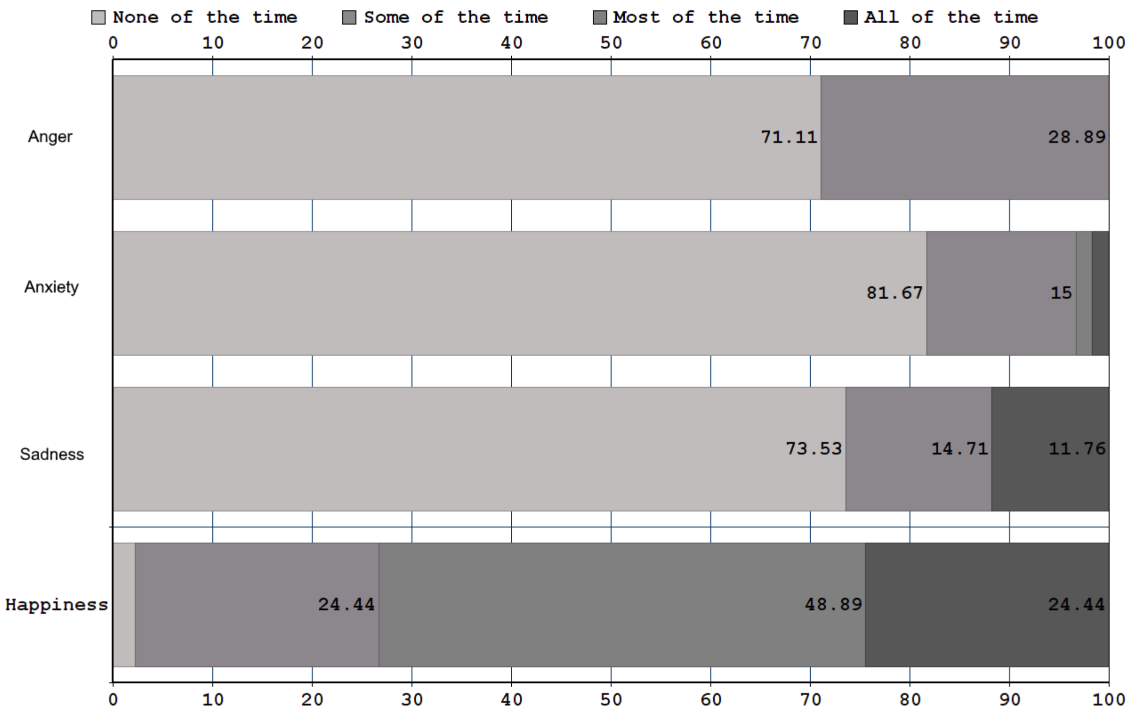

Figure 7.4.: Distribution of answers grouped by emotions on the Computer Emotion Scale.

7.4.3. Feedback By Participants

This section describes what the participants suggested via the feedback questionnaire and comments that were made during the usage of the Chat Agent System.

Most of the participant appreciated that the Chatbot could answer questions quite fast, either replying that it does not have an answer of providing multiple answers to the users as seen in table 7.13. Since the result of a user query can be a multiple choice option, the participants appreciated that they could get options and did not have to think how to make the question more precise in order to get the correct answer. The participants like that the Chat Agent System was able to execute functionalities of three separated systems and provide guided information, where they did not have to browse a lot to get the information.

One of the main complaints of the participants is that they did not get any instructional messages at the start of the use of the chatbot, this made them ask random questions that had nothing to do with the tasks. The improvement of the interface was suggested in order to make it more usable and attractive. Another interesting feature that the users would like to have is voice recognition and voice generation. In the case that the chatbot cannot provide an answer to a query, the participant suggested that they are redirected to other pages with more information or to a real life agent.

Most of the participants (75%) prefere the use the Chat Agent System than the smaXtec system for the execution of the specific tasks they were assigned and 50% of them agreed that it was easier to find answers via the Chat Agent System.

User Feedback
Getting the stuff from the messenger is cool
The answers came quick and clear.
It was easy to use and very intuitive
The anwsers are super fast. Clear and easy
It understood almos all of my queries. It gave clear and concise answers to my questions. I didn't have to hunt for any information.
If I typed in something unclear smaXbot gave a suggestion what it could mean
It was quite easy to use. When I didn't enter a question that could be found, it immideatly provided me suggestions.
The chatbot provided answers quick
Easy to use to find answers to specific questions
Fast answers and guided infromation

Table 7.13.: User Feedback on the Chat Agent System

7.4.4. Analysis of Participant Behavior

Besides the standard scales and user feedback, the analysis of the behavior of participants has been conducted. Listing 7.1 shows a sample conversation between a user and the Chat Agent System, while the participant is executing tasks from table 7.1. In average the participants needed 11.06 interactions with the Chat Agent System in order to get answers to 6 questions.

```

Participant: Hello
Chat Agent System: Hi, how can I help you?

Participant: What is smaXtec?
Chat Agent System: smaXtec offers a comprehensive solution for
    monitoring your herd. Access to important information about heat,
     calving, health and feeding enables you to take the right
    decisions at the right time and improve animal well-being as well
     as farm efficiency


Participant: What is a bolus?
Chat Agent System: The smaXtec Bolus is a measurement device that
    measures activity, temperature
and pH values of Cows.


Participant: When is my data available?
Chat Agent System: Did you mean "How long does it take until the
    first data is available?"?


Participant: How long does it take until the first data is available
    ?
Chat Agent System: After smaXtec Classic/pH Plus Boli have been
    activated and data has been read-out by the smaXtec Base Station,
     it takes at least 30 minutes until data is available for the
    first time in the smaXtec Messenger.


Participant: Does the smaXtec base station need internet?
Chat Agent System: Yes, a permanent connection to the internet is
    needed.
```

Listing 7.1: Sample Conversation with the Chat Agent System

The analysis of participant logs with the Chat Agent System has revealed that the participants tried to ask the Chat Agent System to provide answers in a way that the questions were exactly the same as the task formulation from table 7.1, which reduced the number of interactions between the Chat Agent System and the participants. A small number of the participants, tried to reformulate the questions needed to finish the tasks, which caused that some participants have more than 11.06 interactions with the Chat Agent System. The distribution of interactions with the Chat Agent System per participant can be seen in figure 7.5.

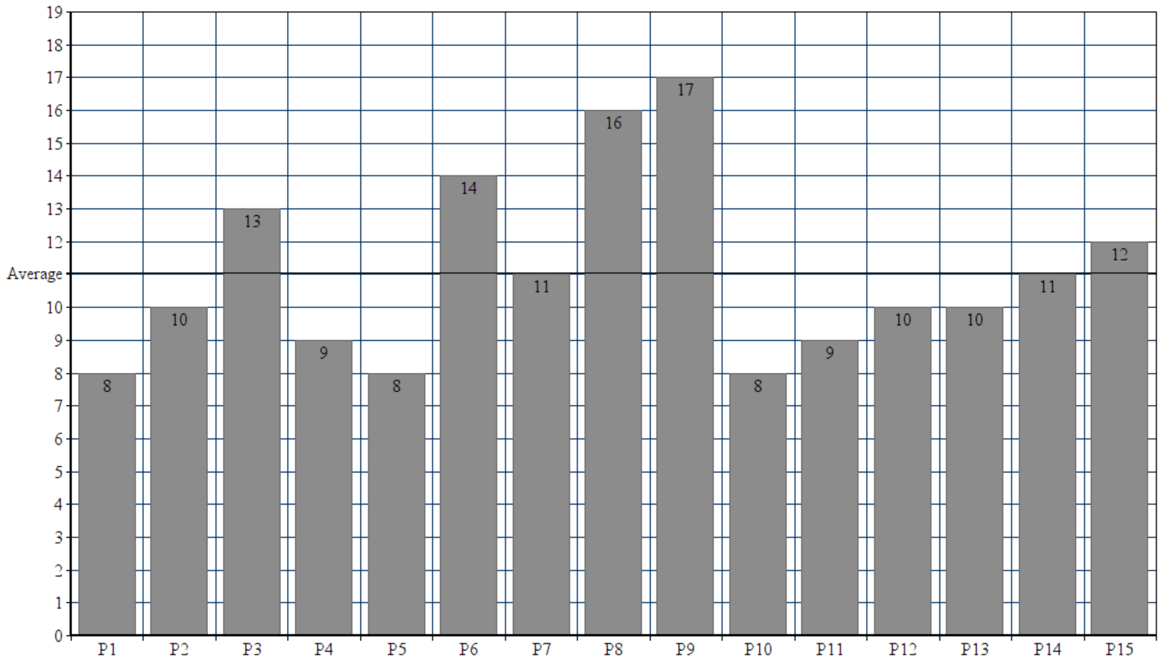

Figure 7.5.: Number of Interactions between the Participants and the Chat Agent System and the Average Number of Interactions.

7.4.5. Limitations

Due to the limitation in the number of participants it is not possible to draw a final conclusion on how well the Chat Agent System could replace or improve the existing smaXtec system.

Since the Chat Agent System was programmed to reply in English and all of the participants did not speak English as their native language, the results might have been influenced by this language barrier.

Considering that the number of participants was only fifteen, where five of the participants were not smaXtec employees, not many differences between the two groups could be extracted. It is visible that the average time needed to find information via the smaXtec system is much higher for the non smaXtec employees. The average execution time for smaXtec employees with the smaXtec system was 6 minutes and 2 seconds, while the non smaXtec employees needed 8 minutes and 43 seconds. When using the Chat Agent System the average execution time for the tasks by the smaXtec employees was 4 minutes and 56 seconds, while the non smaXtec employees needed 4 minutes and 35 seconds.

7.5. Summary

The result of the user survey showed that the participants perceived the Chat Agent System as a positive improvement to the existing system and an interesting tool to use for information retrieval. According to the CES the participant mostly felt happiness while using the Chat Agent System together with excitement and curiosity, while expressing anger in cases when the Chat Agent System was not able to understand their intentions. Based on the difficulty rating of the tasks it is visible that the users executed the tasks with more ease when using the Chat Agent System. The answers provided by the Chat Agent System were more clear to the participants than the answers provided by the smaXtec system. The SUS results determined that the system did not meet the minimum points in order to be declared as a usable system, even though the system was quite close to the limit with 59.92 points in average.

The participants were satisfied with the functionalities that the Chat Agent System brought to them and were able to execute all tasks with ease. The appearance of the Chat Agent System was criticized by the participants, but was perceived with positivity.

The user feedback revealed that there is a lot of space for improvement in the user interface, the way that the users communicate with the system and formulation of the answers that the system provides.

8. Lessons Learned

The focus of this chapter is on the learned lessons during the implementation of the thesis. These lessons learned include details about the literature and implementation process together with the evaluation and results of the evaluation.

8.1. Literature Survey and Implementation

The literature survey in the area of Chatbots and Dialog Systems proved to be quite difficult because there is no unique definition for terms like chatbot or dialog system. According to multiple sources, these terms are equivalent yet other sources distinguish between dialog systems and chatbot. The result of the literature survey concludes that the term Conversation Agents can be used as the parent term for both chatbots and dialog systems.

Since the topic of Conversation Agents is a new field of research and also a profitable field of business, most of the publications are not up to date with existing technologies and more reliable methods. Since a great amount of breakthroughs in this area is done either by companies or private individuals it is hard to access the results of this research since the participants of these breakthroughs do not care about sharing their research. Companies want to keep their results secret in order to be profitable and individuals either do not have time to or do not want to write research work.

Since the creation of conversation agents, the methods have not evolved that fast in comparison with other areas in the field of Computer Science. The huge amount of data that is produced daily by humans and the evergrowing improvements in processing power have lead Conversation Agents to popularity nowadays. The new age processing speed and the appearance of systems that contain huge amounts of information, have lead to the development of natural language processing and natural language understanding techniques that are based on pattern matching and machine learning methods. These new methods improve the workings of dialog systems considerably.

Even though the latest technologies were used to create the prototype, they were still not able to predict the intent of users with high precision. Because the input of each user is different, it was hard to create patterns for recognition of the inputs that would work with all users. It would have been better to base the patterns on existing user input than to create patterns based on provided answers and expert knowledge. Even in this case due to the high degree of freedom when creating user queries, there would not be one pattern that matches all variations.

The prototype was implemented to understand and interpret user queries in the English language, which was easier to implement than any other language because of the ease of access to material, guides, data and patterns in the English language.

Combining multiple natural language processing and dialog manager technologies was needed during the implementation which led to an increase of complexity in the Chat Agent System, because these components needed to communicate in order for the whole system to work. It would have been preferable to use one technology to implement all of the requirements, it was not possible during the creation of the Chat Agent System.

8.2. Outcome

Due to the limited number of participants and that the majority of participants were employees at smaXtec, the results of the user survey were heavily influenced by the user's familiarity with the existing system. For further studies, it would be beneficial to have a bigger group of individuals where the distribution of employees and non-employees is evenly balanced.

Since the prototype was implemented in a way that there were no quality or usability tests in different periods of the implementation, the Chat Agent System was created with certain problems that could have been detected in earlier stages and improved during the implementation. The use of a agile development process in combination with constant user tests would have led to the creation of a better prototype, but would have needed more time.

9. Conclusion and Future Work

Ideas for future research projects together with the summarization of the research questions and goals of the thesis are explained in this chapter.

9.1. Conclusion

The goal of the company smaXtec was to determine if a dialog system can be integrated into their existing system and if it could be a cost-effective system that can handle the increased number of customer requests while not reducing the quality of the service they provide to the customers. The thesis prototype was created to determine if the goals of the company were possible. To determine if it was possible to implement such a system the thesis focused to provide insights in the following research questions:

1. Can a dialog system be used as an information retrieval system and a replacement for frequently asked questions pages?
2. Would the users utilize a dialog system to retrieve data and acquire answers to specific questions rather than an existing web based system?
3. Does the dialog system increase the speed of question-answering in comparison with the existing system?
4. Do the users feel comfortable with the usage of the dialog system?
5. Which emotions do they experience when using the dialog system?

To provide answers to these questions, following the implementation of the Chat Agent System, a user study was conducted. The user study focused to determine if the created system was usable and provided an adequate proof of concept that a Dialog System can be used in the role of an information retrieval system and a customer support system.

Based on the user study it was concluded that a dialog system can be used as an information retrieval system and a replacement for frequently asked questions pages. The main problem that makes dialog systems not appealing in these roles is that there exist a huge number of languages with different rules and that is hard to adapt dialog systems to work with them all. Besides this, due to the hard to predict nature of humans, the user input can sometimes variate largely from the expected input for these systems. This increases the complexity of even simple user queries.

The majority of the participants (73%) said that they preferred the Chat Agent System rather than the smaXtec system.

The System Usability Scale (SUS) had the average result of 78.66 points which classified the Chat Agent System as a usable system by the participants. With the use of the Chat Agent System the time needed to execute specific tasks, which ranged from information retrieval to finding answers to frequently asked questions, was reduced in comparison to the existing smaXtec system.

Based on the Computer Emotion Scale (CES), Happiness and Anger were the dominant emotions while Curiosity and Satisfaction were the prevailing feelings that the participants felt when using the Chat Agent System. This in simple terms means that the participants felt quite comfortable when using the Chat Agent System.

9.2. Future Work

Many topics that were discussed in the thesis can be used as a base for further development and research projects.

Future research projects could include a more personalized CA that is user-dependent, where the CA learns about specific users and provides more personalized answers to questions. Besides personalized answering, another research topic could focus on a more proactive system, where the CA notifies the user about tasks that he/she needs to execute.

Automatic topic creation or question-answer pairing might be an interesting direction for future work. The utilization of machine learning tools and frameworks for advance an administration panel is definitely an interesting field of research. Advance methods that guide and help administrators improve the CA system have high research potential.

Even though the CA is capable of understanding user queries only basic NLP concepts have been used in this thesis. Implementing advanced NLP techniques on to the existing CA base might lead to interesting results.

Besides research projects oriented on the improvement of the CA, better evaluation of the system could also be a goal of future projects. Determining what are concrete negative and positive sides of CA based customer support and information retrieval systems is an important topic for future research. A comparison of question-answering performance between a real-life customer support agent and the CA system could result in interesting findings.

Appendix

Appendix A.

Feedback Questionnaire

Feedback Questionnaire

Did you use the smaXtec system earlier?	Yes		No		
The Chatbot was easy to use.	1	2	3	4	5
The Chatbot improved the experience of the system.	1	2	3	4	5
It was easier to find answers to questions via the chatbot.	1	2	3	4	5
What did you like about the chatbot?					
What did you dislike about the chatbot?					
Do you have any suggestions for improvements?					
Do you have any functionalities that you would like to add?					
Do you prefere the chatbot or the smaXtec system?					

1 – Not at All; 5 – Fully Agree

Figure A.1.: Feedback Questionnaire

References

Abdulghani, T., Najar, M. A., Belaroussi, R., Mothe, J., Ryzhov, M., & Samoskaite, S. (2018). Browsing information retrieval system results. In *Coria.*

Abdul-Kader, S. A., & Woods, D. J. (2015). Survey on chatbot design techniques in speech conversation systems. *International Journal of Advanced Computer Science and Applications*, *6*(7). Retrieved from `http://dx.doi.org/10.14569/IJACSA.2015.060712`

Abu Shawar, B., & Atwell, E. (2007). Chatbots: are they really useful? *LDV-Forum: Zeitschrift für Computerlinguistik und Sprachtechnologie*, *22*(1), 29–49. doi: 10.1.1.106.1099

AbuShawar, B., & Atwell, E. (2015, 12). Alice chatbot: Trials and outputs. *Computación y Sistemas*, *19*. doi: 10.13053/cys-19-4-2326

Asadi, A. R., & Hemadi, R. (2018, 04). Design and implementation of a chatbot for e-commerce.

Bangor, A., Kortum, P. T., & Miller, J. T. (2009). Determining what individual sus scores mean: adding an adjective rating scale..

Bani, B., & Singh, A. P. (2017, 01). College enquiry chatbot using a.l.i.c.e. *International Journal of New Technology and Research (IJNTR)*, *3*.

Bergner, M., & Johannsen, F. F. (2016). Supporting business process improvement with natural language processing: A model-based approach. In H. C. Mayr & M. Pinzger (Eds.), *Informatik 2016* (p. 717-730). Bonn: Gesellschaft für Informatik e.V.

Bird, S., Klein, E., & Loper, E. (2009). *Natural language processing with python* (1st ed.). O'Reilly Media, Inc.

Bishop, C. M. (2006). *Pattern recognition and machine learning (information science and statistics)*. Berlin, Heidelberg: Springer-Verlag.

Bradeško, L., & Mladenić, D. (2012, 01). A survey of chatbot systems through a loebner prize competition. In (Vol. C, p. 34).

Brandtzaeg, P., & Følstad, A. (2017, 11). Why people use chatbots..

Brooke, J. (1996). *"sus-a quick and dirty usability scale." usability evaluation in industry*. CRC Press. Retrieved from https://www.crcpress.com/product/isbn/9780748404605 (ISBN: 9780748404605)

Bruce, W. (n.d.). *Beyond facade: Pattern matching for natural language applications.* Retrieved from https://www.gamasutra.com/view/feature/134675/beyond_fa%EF%BF%BDade_pattern_matching_.php

Bruce, W. (2008). *Beyond aiml: Chatbots 102.* Retrieved from https://www.gamasutra.com/view/feature/132155/beyond_aiml_chatbots_102.php

Bruce, W. (2010). *Suzette, the most human computer.* Retrieved from https://www.gamasutra.com/blogs/BruceWilcox/20110623/89684/Suzette_the_Most_Human_Computer.php

Bruce Wilcox. (2013). *Chatscript basic user's manual.* Retrieved 2019-06-05, from http://chatscript.sourceforge.net/Documentation/ChatScript%20Basic%20User%20Manual.pdf

CareerBuilder. (2017). *Thirty-six percent of employers lack a structured onboarding process for new employees, according to new careerbuilder survey* (Tech. Rep.). Author.

Chakrabarti, C., & Luge, G. F. (2015). Artificial conversations for customer service chatter bots: Architecture, algorithms, and evaluation metrics.

Chen, I. J., & Popovich, K. (2003). Understanding customer relationship management (crm): People, process and technology. *Business Process Management Journal*, *9*(5), 672-688. Retrieved from https://doi.org/10.1108/14637150310496758 doi: 10.1108/14637150310496758

Codd, E. F. (1990). The relational model for database management, version 2..

Colby, W., & Gilbert. (1966). A computer method for psychotherapy. Journal of Nervous and Mental Disease.

Couto, J. (2017). *Building a Chatbot: analysis & limitations of modern platforms.*

C.P. Shabariram, C. I., V.Srinath, & Vidhya, M. (2017, 03). Ratatta: Chatbot application using expert system. *International Journal of Advanced Research in Computer Science and Software Engineering*, *7*.

Csaky, R. (2019). Deep learning based chatbot models. *ArXiv*, *abs/1908.08835*.

Cui, L., Huang, S., Wei, F., Tan, C., Duan, C., & Zhou, M. (2017, 01). Superagent: A customer service chatbot for e-commerce websites. In (p. 97-102). doi: 10.18653/v1/P17-4017

Dabbagh, M., Peck Lee, S., & Parizi, R. (2015, 07). Functional and non-functional requirements prioritization: empirical evaluation of ipa, ahp-based, and ham-based approaches. *Soft Computing*. doi: 10.1007/s00500-015-1760-z

Dakkak, O., Ghneim, N., & Salman, I. (2014, 10). An interactive speech web site in arabic and english. *Asian Journal of Information Technology*, *13*. doi: 10.3923/ajit.2014.720.724

della Cava, M. (2016). *Usa today: Microsoft ceo nadella: "bots are the new apps"*. Retrieved 2016-03-30, from https://eu.usatoday.com/story/tech/news/2016/03/30/microsof-ceo-nadella-bots-new-apps/82431672

Deshpande, A. U., Shahane, A., Gadre, D., Deshpande, M. S., & Joshi, P. M. (2017). A survey of various chatbot implementation techniques..

Deshpande, S. A. G. D. D. M., Aditya, & Joshi, P. M. (2017). A SURVEY OF VARIOUS CHATBOT IMPLEMENTATION TECHNIQUES .

Ellatif, M. M. A. (2008, 01). A cluster technique to evaluate effect of ecrm on customers' satisfaction of e-commerce websites. , 10.

Eurostat. (2016). *European union labour force survey* (Tech. Rep.). Author.

Freeman, A. (2014). *Pro angularjs* (1st ed.). Berkely, CA, USA: Apress.

Goodman, J., & Wilke, M. (2007). *Icmi customer management insight magazine* [Online Journal].

Hogeveen, H., Rutten, N., Kamphuis, C., & Voort, M. (2017, 01). Economics of precision dairy monitoring techniques.

Horvitz, E. (1999, September). Mixed-initiative interaction. *IEEE Intelligent Systems*, 14-24. Retrieved from https://www.microsoft.com/en-us/research/publication/mixed-initiative-interaction/

Initiative, A. (2017). *Chatbots in customer service* [Online Journal].

Janarthanam, S. (2017). *Hands-on chatbots and conversational ui development: Build chatbots and voice user interfaces with chatfuel, dialogflow, microsoft bot framework, twilio, and alexa skills*. Packt Publishing.

John Hutchins, W. (2004, 09). The georgetown-ibm experiment demonstrated in january 1954. In (Vol. 3265, p. 102-114). doi: 10.1007/978-3-540-30194-3_12

Jurafsky, D., & Martin, J. H. (2018). *Speech and Language Processing*.

Kates, C. (1976, 01). A critique of chomsky's theory of grammatical competence. *Forum Linguisticum*, *I*, 15-24.

Kay, R. H., & Loverock, S. (2008). Assessing emotions related to learning new software: The computer emotion scale. *Computers in Human Behavior*, *24*, 1605-1623.

Kerly, A., Hall, P., & Bull, S. (2007). Bringing chatbots into education: Towards natural language negotiation of open learner models. *Knowledge-Based Systems*, *20*(2), 177–185. doi: 10.1016/j.knosys.2006.11.014

Khurana, D., Koli, A., Khatter, K., & Singh, S. (2017, 08). Natural language processing: State of the art, current trends and challenges.

Kluwer, T. (2011). From Chatbots to Dialog Systems. *Conversational Agents and Natural Language Interaction: Techniques and Effective Practices*, 1. Retrieved from `http://www.igi-global.com/viewtitlesample.aspx?id=54631` doi: 10.4018/978-1-60960-617-6.ch001

Kruschwitz, U., De Roeck, A., Scott, P., Steel, S., Turner, R., & Webb, N. (1999, 12). Natural language access to yellow pages. In (p. 34-37). doi: 10.1109/KES.1999.820113

Lasek, M., & Jessa, S. (2013). Chatbots for customer service on hotels' websites. *Information Systems in Management*, *2*(2), 146–158.

Le, Q. H., Xie, J., Millington, D., & Waniss, A. (2015, 12). Comparative performance analysis of postgresql high availability database clusters through containment. *International Journal of Advanced Research in Computer and Communication Engineering*, *4*, 526-533. doi: 10.17148/IJARCCE.2015.412150

Loper, E., & Bird, S. (2002). Nltk: The natural language toolkit. In *In proceedings of the acl workshop on effective tools and methodologies for teaching natural language processing and computational linguistics. philadelphia: Association for computational linguistics.*

López, G., Quesada, L., & Guerrero, L. A. (2018). Alexa vs. siri vs. cortana vs. google assistant: A comparison of speech-based natural user interfaces. In I. L. Nunes (Ed.), *Advances in human factors and systems interaction* (pp. 241–250). Cham: Springer International Publishing.

Margallo, J. A. S. (2007). $Test_de_turing.jpg.Retrieved2007-05-20, from$

McCartney, S. (1999). *Eniac: The triumphs and tragedies of the world's first computer*. Walker & Company.

Mislevics, A., Grundspenkis, J., & Rollande, R. (2018). A systematic approach to implementing chatbots in organizations - rtu leo showcase. In *Bir workshops.*

Molly, M. (2017, 10). *How messenger bots and manychat took molly mahoneys facebook live video reach past 1 million.* Retrieved from https://manychat.com/blog/how-molly-mahoney-uses-manychat-and-facebook-live/

Murray, K. (1991, 01). A test of services marketing theory: Consumer information acquisition activities. *Journal of Marketing*, *55*. doi: 10.2307/1252200

My Clever Agency. (2018). *Chat bots! a consumer research study.* Retrieved 2019-06-05, from http://www.mycleveragency.com/newsletters/view/ffcca7778d

Nielsen, N. (2011). *State of the media: The social media report* (Tech. Rep.). nielsen.

Pappu, A., & Rudnicky, A. (2014, June). Knowledge acquisition strategies for goal-oriented dialog systems. In *Proceedings of the 15th annual meeting of the special interest group on discourse and dialogue (SIGDIAL)* (pp. 194–198). Philadelphia, PA, U.S.A.: Association for Computational Linguistics. Retrieved from https://www.aclweb.org/anthology/W14-4326 doi: 10.3115/v1/W14-4326

Radlinski, F., & Craswell, N. (2017, March). A theoretical framework for conversational search. In *Chiir 2017* (CHIIR 2017 ed.). ACM. Retrieved from https://www.microsoft.com/en-us/research/publication/theoretical-framework-conversational-search/

Radziwill, N., & Benton, M. (2017, 04). Evaluating quality of chatbots and intelligent conversational agents.

Risueño, T. (n.d.). *How we trained our chatbot with one billion sentences in 2 days.* Retrieved from https://chatbotslife.com/how-we-trained-our-chatbot-with-one-billion-sentences-in-2-days-cffbdd06b505

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65–386.

R. Ranoliya, B., Raghuwanshi, N., & Singh, S. (2017, 09). Chatbot for university related faqs. In (p. 1525-1530). doi: 10.1109/ICACCI.2017.8126057

Samsudin, W., & Juhary, A. (2014, 05). The evolution of relationship marketing (rm) towards customer relationship management (crm): A step towards company sustainability.

Savoy, J., & Gaussier, E. (2010, 01). Information retrieval.

Schneider, C. (2017). *10 reasons why ai-powered, automated customer service is the future.* Retrieved 2017-10-16, from https://www.ibm.com/blogs/watson/2017/10/10-reasons-ai-powered-automated-customer-service-future/

Shah, H., Warwick, K., Vallverdú, J., & Wu, D. (2016). Can machines talk? Comparison of Eliza with modern dialogue systems. *Computers in Human Behavior*, *58*, 278–295. doi: 10.1016/j.chb.2016.01.004

Shalev-Shwartz, S., & Ben-David, S. (2014). *Understanding machine learning: From theory to algorithms*. New York, NY, USA: Cambridge University Press.

Sharma, G., & Malik. (2017). An Intelligent Behaviour Shown by Chatbot System.

Shawar, B., & Atwell, E. (2002). A comparison between ALICE and Elizabeth chatbot systems. *Computer*(December). Retrieved from `http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.3.4703&rep=rep1&type=pdf`

Shawar, B. A., Atwell, E., & Roberts, A. (2005). Faqchat as in information retrieval system..

Steven Bird, E. K., & Loper, E. (2009). *Natural Language Processing with Python*.

Sutton, R. S., & Barto, A. G. (1998). Reinforcement learning: An introduction. *IEEE Trans. Neural Networks*, *9*(5), 1054–1054.

Tiha, A. (2018). *Intelligent chatbot using deep learning* (Doctoral dissertation). doi: 10.13140/RG.2.2.14006.75841

Turing, A. (1950). COMPUTING MACHINERY AND INTELLIGENCE. *Mind*, 22.

Valtolina, S., Barricelli, B. R., Gaetano, S. D., & Diliberto, P. (2018). Chatbots and conversational interfaces: Three domains of use. In *Copda@avi.*

van Woudenberg, A. (2014). *A Chatbot Dialogue Manager Chatbots and Dialogue Systems: A Hybrid Approach* (Unpublished doctoral dissertation). Open University of the Netherlands.

Walton, F. (2016). *Twitter bootstrap: For starters*. USA: CreateSpace Independent Publishing Platform.

Weizenbaum, J. (1966). ELIZA — A Computer Program For the Study of Natural Language Communication Between Man And Machine. *Communications of the ACM*, *9*(1), 36–45. Retrieved from `http://joi.jlc.jst.go.jp/JST.Journalarchive/jje1965/2.3_1?from=CrossRef` doi: 10.5100/jje.2.3{_}1

Wiederhold, G., & McCarthy, J. (1992, 06). Arthur samuel: Pioneer in machine learning. *IBM Journal of Research and Development*, *36*, 329 - 331. doi: 10.1147/rd.363.0329

Wikimedia. (2018). $eliza_title.jpg.Retrieved2018-07-05,from$

Xu, A., Liu, Z., Guo, Y., Sinha, V., & Akkiraju, R. (2017, 05). A new chatbot for customer service on social media.
doi: 10.1145/3025453.3025496

Yan, Z., Duan, N., Bao, J., Chen, P., Zhou, M., Li, Z., & Zhou, J. (2016, August). DocChat: An information retrieval approach for chatbot engines using unstructured documents. In *Proceedings of the 54th annual meeting of the association for computational linguistics (volume 1: Long papers)* (pp. 516–525). Berlin, Germany: Association for Computational Linguistics. Retrieved from https://www.aclweb.org/anthology/P16-1049 doi: 10.18653/v1/P16-1049

Zimmermann, P. (2019). *Conversational ui: How chatbots change customer communication.* Retrieved 2019-06-05, from https://innovation-implemented.com/en/conversational-ui-so-veraendern-chatbots-die-kundenkommunikation/