Bachelor Thesis

November 10, 2021

Designing a Chatbot

An investigation for the University of Zurich

Eleonora Pura

of Locarno, Switzerland (17-732-678)

supervised by

Prof. Dr. Harald C. Gall

Dr. Pasquale Salza

University of Zurich^{UZH}

s.e.a.l. software evolution & architecture lab

Bachelor Thesis

Designing a Chatbot

An investigation for the University of Zurich

Eleonora Pura

University of Zurich UZH

s.e.a.l. software evolution & architecture lab

Bachelor Thesis

Author: Eleonora Pura, eleonora.pura@uzh.ch

URL: https://github.com/epura/Bachelor_thesis

Project period: 10.5.2021 - 10.11.2021

Software Evolution & Architecture Lab
Department of Informatics, University of Zurich

Acknowledgements

This Bachelor Thesis was supported by the software evolution & architecture lab (s.e.a.l.) group of the University of Zurich. I would like to thank Prof. Dr. Harald C. Gall for this possibility.

Furthermore, I would like to thank Dr. Pasquale Salza and Marco Edoardo Palma for their supervision and support. I would like to thank them for their availability, the meetings and valuable suggestions. A special thank you to Marco Edoardo Palma for taking the time to correct my thesis and give me ideas for improvement.

Last but not least, I want to thank the IfI academic coordinator Nathan Labhart and administrative assistant Daniela Bärtschi for their willingness to conduct an interview for constructing the use case.

Abstract

This thesis presents a use case created with the collaboration of the Faculty of Informatics (IfI) of the University of Zurich. Their student office interacts with students on a daily basis through emails, phone calls, and one-site meetings. However, many of their interactions are repetitive or on topics that do not fall under their responsibility. These aspects could be improved thanks to the creation of a chatbot, which would allow them to take care of this type of interactions in a more structured and faster way. Chatbots are systems capable of simulating a natural language conversation with a human. Their responses can be generated either on the basis of predefined rules or with the help of machine learning approaches. Chatbots can be useful in different areas such as entertainment, information retrieval, and e-commerce. The aim of this thesis is on one hand to find potential areas of interest for which a chatbot could help improve the efficiency of the interactions between the student office and the students. But most importantly, it serves as a design exercise to see how a chatbot can replace these areas. After conducting an interview and its subsequent in-depth analysis, it was possible to better understand what the main activities of the student office are. This allowed the application of a Software Engineering process, for which the first step was the definition of the requirements, and the second the conception of a first ad-hoc design for a chatbot. Finally, some suggestions for practical solutions are presented. The final result is a complete design of a rule-based chatbot, able to answer questions about the writing process of Bachelor and Master theses and other important topics.

Zusammenfassung

Diese Arbeit stellt einen Anwendungsfall vor, der in Zusammenarbeit mit der Fakultät für Informatik (IFI) der Universität Zürich erstellt wurde. Deren Studentenbüro interagiert täglich mit Studenten durch E-Mails, Telefonanrufe und Treffen vor Ort. Viele dieser Interaktionen wiederholen sich jedoch oder betreffen Themen, die nicht in ihren Zuständigkeitsbereich fallen. Diese Aspekte könnten durch die Entwicklung eines Chatbots verbessert werden, welcher diese Art von Interaktionen strukturierter und schneller ermöglichen würde. Chatbots sind Systeme, die in der Lage sind, ein Gespräch in natürlicher Sprache mit einem Menschen zu simulieren. Ihre Antworten können entweder auf der Grundlage vordefinierter Regeln oder mit Hilfe von Machine Learning Ansätzen generiert werden. Chatbots können in verschiedenen Bereichen wie Entertainment, der Informationsbeschaffung und dem elektronischen Handel nützlich sein. Ziel dieser Arbeit ist es zum einen, potentielle Interessensgebiete zu finden, in denen ein Chatbot helfen könnte, die Interaktion zwischen dem Studentenbüro und den Studenten effizienter zu gestalten. Vor allem aber dient sie als Designübung, um zu beobachten, wie ein Chatbot diese Bereiche ersetzen kann. Nach der Durchführung eines Interviews und der anschliessenden eingehenden Analyse war es möglich, die Hauptaktivitäten des Studentenbüros besser zu verstehen. Dies ermöglichte die Anwendung eines Software-Engineering-Prozesses, bei dem der erste Schritt die Definition der Anforderungen und der zweite die Gestaltung eines ersten Ad-hoc-Designs für einen Chatbot war. Schliesslich werden einige Vorschläge für praktische Lösungen vorgestellt. Das Endergebnis ist ein vollständiger Entwurf eines regelbasierten Chatbots, der nützlich ist, um Fragen über den Schreibprozess von Bachelor- und Masterarbeiten, sowie über weitere wichtige Themen zu beantworten.

Contents

1 Introduction **1**

2 Chatbots **3**

2.1 History . . . 3

2.2 Classification . . . 4

2.2.1 Knowledge Domain . . . 4

2.2.2 Goal . . . 6

2.2.3 Service Provided . . . 7

2.2.4 Interaction Mode . . . 7

2.2.5 Response Generation . . . 8

2.3 General Architecture . . . 9

2.4 Implementation . . . 10

2.5 Applications: Companies . . . 11

2.5.1 Example: Insurances . . . 12

2.6 Opportunities and Challenges . . . 12

3 Use Case **15**

3.1 Presentation . . . 15

3.2 Interview . . . 15

3.3 Scope Definition . . . 17

3.4 Requirements . . . 17

3.4.1 User Requirements . . . 17

3.4.2 Functional Requirements . . . 18

3.4.3 Non-functional Requirements . . . 19

3.5 Design . . . 19

3.5.1 Decisions . . . 19

3.5.2 Functioning . . . 25

3.5.3 Practical Solutions . . . 26

3.5.4 Mockups . . . 27

4 Conclusion **31**

A Requirements **33**

A.1 User Requirements . . . 33

A.1.1 FAQ . . . 33

A.1.2 Thesis Procedure . . . 33

A.2 Functional Requirements . . . 34

A.2.1 FAQ 34
A.2.2 Thesis Procedure 34
A.3 Non-functional Requirements 35

B Interview to Requirements and Design Decisions 37

List of Figures

2.1 Classification of chatbots based on [35], [2] and [22] 5
2.2 Own representation of the general architecture of a chatbot based on [33] and [27] 10

3.1 Areas covered by the use case chatbot 18
3.2 Difference between CSV, XML and JSON file formats 22
3.3 Web Scraping Architecture 23
3.4 Idea of internal functioning of the IfI chatbot based on [46] 26
3.5 Example of a possible successful conversation with the IfI chatbot 28
3.6 Example of a possible unsuccessful conversation with the IfI chatbot 29

List of Tables

2.1 Own representation of an example that shows the functioning of the chatbot architecture based on [27] 10
2.2 Representation of identified opportunities and challenges with corresponding literature 14

3.1 Chatbot classification for the IfI use case 21
3.2 Possible practical solutions for the IfI chatbot implementation 27

List of Listings

3.1 Function that finds the links of the files in the While Studying page 23

Chapter 1

Introduction

A chatbot, short for chatterbot, can be defined as a computer program which is able to simulate a conversation with a human being in natural language. This can happen through text chats, voice commands, or both. These conversational agents use artificial intelligence (AI) and can operate either using machine learning techniques or be based on a set of rules [18].

Many studies have already contributed in proving their usefulness. Shawar and Atwell identified and analysed many interesting applications for chatbots such as in education, information retrieval and entertainment [47], but also e-commerce and business. Research has shown that users find chatbots easy to use, capable of understanding their needs and able to support them with little effort and interaction time [47].

Chatbots are becoming increasingly widespread and are also making their way into the market with vendors such as IBM, Amazon or Google. As a consequence, they are progressively becoming part of enterprises' reality: suffice it to say that nowadays chatbots, such as XOR [63], are also used as job recruiters. One of the main advantages of chatbots is that conversations are slower and structured, and consequently can capture information that would otherwise be missed in one-shot interactions.

When a chatbot needs to be created for a specific case, it is necessary to design it and adapt it according to the characteristics of the given context. Thanks to a practical case created with a collaboration with the Department of Informatics of the University of Zurich (IfI), this challenge will be addressed in this thesis. Using the activities of their student office as a starting point, it will be demonstrated how a chatbot can be useful to manage entire activities, or at least act as a first filter capable of reducing the workload of the staff. Current interactions of the student office are mainly with students and include on-site meetings, emails and phone calls. An interview with two department administration workers helped identify suitable activities that can be improved using a chatbot system. The relevant areas that were identified are the Bachelor and Master thesis process, the graduation process, and the indication of the contact information and responsibilities of each department. Software Engineering processes were applied in order to convert the interview's findings into a set of formal requirements, from which a design for an IfI chatbot could be derived. It can be stated that a chatbot can help to improve the efficiency of the the student office interactions, as chatbots in similar applications have already proved to be effective in studies such as [31], [41] and [10].

Chapter 2 starts with a discussion over various aspects of chatbots. This background research is required in order to list the necessary findings for the carrying out of a well informed design project in chapter 3. The contents of this theoretical part include chatbots' history, general classification, architecture and implementation. Successively, chatbots applications such as their use in a business environment are going to be discussed. To conclude the chapter, a reflection on the opportunities and challenges faced by chatbots will be made.

Chapter 3 presents the IfI practical use case. This includes the use case presentation, the interview analysis, the requirements specification and the formalisation of all the design decisions that have been made. The final outcome is a rule-based chatbot for which the set of responses is half manually created and half completed with dynamic variables extracted through Web Scraping.

Chapter 2

Chatbots

This chapter serves as an introduction to the world of chatbots. As a first step, it is necessary to know where they come from and how they have developed over the years (2.1). Subsequently, some of their facets will be shown thanks to a classification model presented in 2.2. In order to understand their inner workings, a basic architecture diagram is presented in section 2.3. After that, some implementation methods that can be used for their creation are shown in section 2.4. Section 2.5 is a reflection on how nowadays chatbots can be used as an enterprise resource. This will be supported by the example of the insurance sector (2.5.1). Finally, some of the opportunities and challenges chatbots face are presented in section 2.6.

2.1 History

It was only 1950 when Alan Turing wondered whether machines were able to think. Precisely, if a human could communicate with a machine, without noticing that he was not interacting with a real person. This concept was called "the imitation game", and is generally considered to be the forerunner of what chatbots have become today [54].

The first real chatbot, ELIZA, appeared sixteen years later. This was the first software application that was able to conduct conversations in natural language. The successful script put the chatbot in the shoes of a psychotherapist. This utilised a rule-based method for the extraction of a predefined set of keywords, from which "intelligent" responses were derived thanks to the use of a transformation rule. The first n keywords were selected thanks to an internal ranking system.

One of the main strengths of ELIZA was that the script was not directly part of the program, because it relied on data. This allowed the program to potentially use different recognition patterns and responses and to not be limited to a specific language [59]. On the other hand, it could only talk about a limited set of subjects, it was not suitable for long discussions and could not collect any information about the context from the conversation [2].

In the early Seventies, PARRY was created. PARRY can be seen as the counterpart of ELIZA, as it behaved like a paranoid patient [13]. Despite this being one of the first important attempts at creating a chatbot, it still had several shortcomings. These included the ability to interpret emotions and language, the long response times and the much needed ability to learn from previous interactions [2].

Jabberwacky is an AI chatbot developed for entertainment, companionship and communication. It was first conceived in 1988, and went online in 1997. The system was able to learn from its users by adding their input to a linguistic database and, with the help of contextual pattern matching techniques, carefully choose its output. Responses were therefore neither hard coded nor language dependent, and always relied on past conversations [11, 14].

A.L.I.C.E. (Artificial Linguistic Internet Computer Entity) was first implemented by Richard Wallace in 1995 and was the first chatbot based on AIML (Artificial Intelligence Markup Language), which utilized XML language for generating chatbots able to create responses as a consequence of received stimulus. Throughout the years it was greatly improved. For example, thanks to the rise of the web it was able to collect a large scale of natural language data. This resulted in A.L.I.C.E. winning the Loebner Prize at the Turing Test contest for the category "the most human computer" three times: in 2000, 2001 and 2004 [57].

In 2001, ActiveBuddy created the chatbot SmarterChild. What made it special was the fact that it was the first chatbot that was used for reasons that went beyond pure entertainment. In fact, SmarterChild was also capable of providing useful information such as sports scores, stock market information and movie quotes, to name a few. With more than 30 million users on AOL and Windows Live Messenger, the chatbot was then bought by Windows in 2006 for the substantial sum of 46 million USD. Today, it is referred to as the precursor of some of the most famous voice assistants such as Apple's Siri and Samsungs' S Voice [25].

Siri was first developed by SRI International and was later integrated into the iPhone, released in 2011 [6,25]. It was a smart personal voice assistant used both for engaging with users in simple small talk conversations, providing them with useful information, or receiving commands. Later on, other tech giants developed their own AI assistants. Google developed its own Google Assistant, once called Google Now [53]. Home assistants also made their way into the market with Amazon Alexa and Google Home.

A significant change occurred when in 2015 Telegram opened its bot platform. Before, companies were developing their own individual chatbots. With this innovation, single developers were allowed to create their own chatbots serving numerous purposes among which were polls, news, games, integration and entertainment. Slack, another communication platform, immediately launched its bot user's platform in the same year. After that many other messaging software followed the trend: from Facebook, who was the most influencial in the rise of chatbots, to Skype, Kik and WeChat [25].

2.2 Classification

Chatbots can be classified according to several criteria. In 2.2.1 a fundamental distinction is going to be briefly presented: the difference between open domain and closed domain chatbots. Another important differentiator is the goal that the conversational agent wants to reach, which can be both task-oriented and non-task-oriented. Within non-task oriented chatbots one can further distinguish between conversation-oriented and information-oriented chatbots [35]. This criterion is discussed in 2.2.2. The service provided by chatbots can be interpersonal, intrapersonal or inter-agent. This is explained in 2.2.3. Different interaction modes for the dialogues include text, voice and images [2]. 2.2.4 will give an introduction on the first two modes. In 2.2.5 the different strategies for generating the chatbot responses are going to be described. The three most common approaches are: rule-based, retrieval-based or generative model [19]. Figure 2.1 gives a clearer picture of the classification.

2.2.1 Knowledge Domain

Open Domain

Open domain chatbots have no specific domain and can therefore get involved with many different topics. Examples of them are Alexa and Siri [35]. Although there are great opportunities

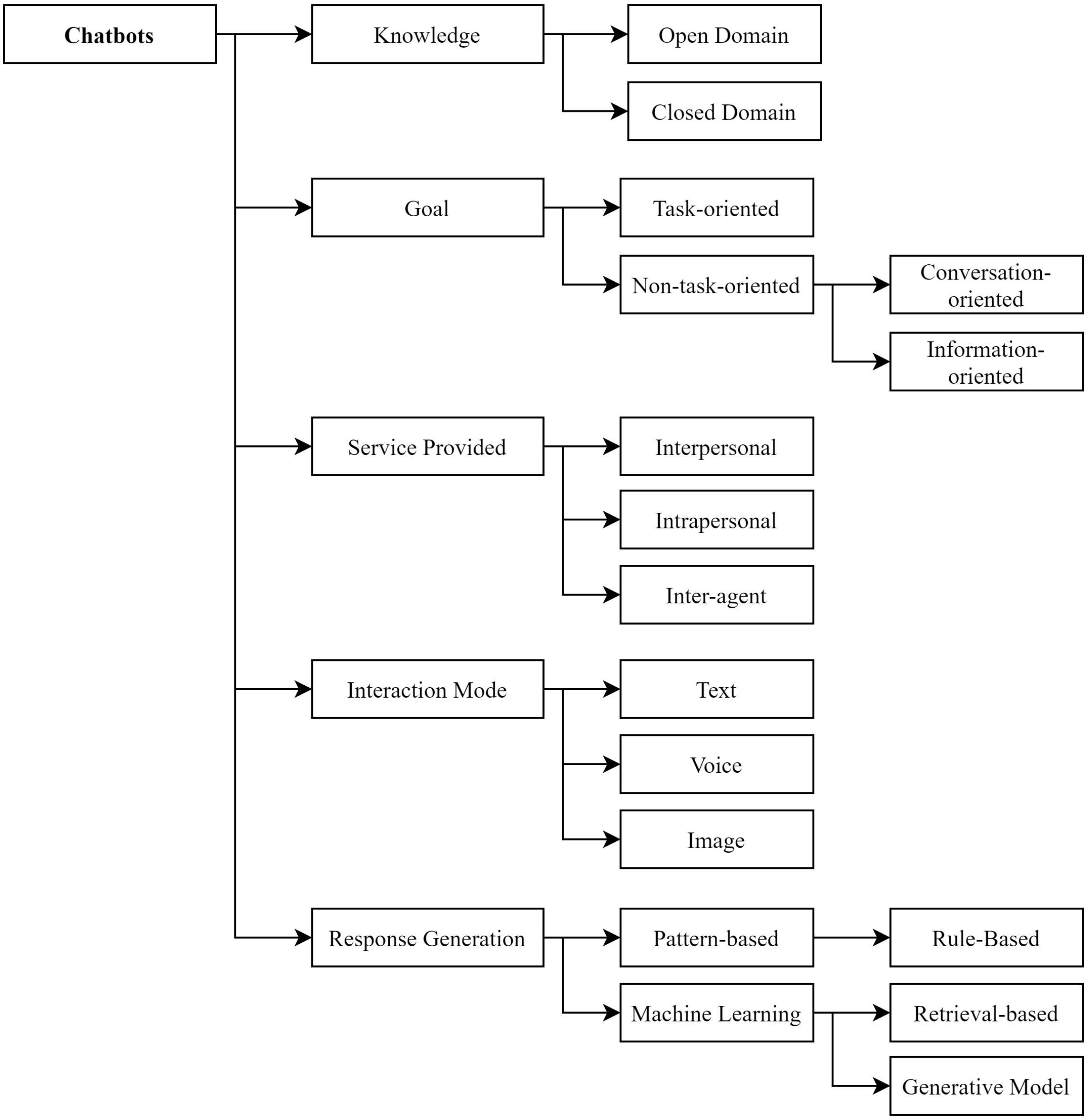


Figure 2.1: Classification of chatbots based on [35], [2] and [22]

behind these types of chatbots, they all have weaknesses due to the often generic and nonsensical responses that are returned to open-ended inputs [4].

Closed Domain

Closed domain chatbots are effective in answering questions about a specific knowledge domain [35] with the help of keywords and intents [4].

More specifically, the two most important Natural Language Understanding (NLU) notions for breaking up a sentence are entities and intents. An entity is an "abstract object" of particular interest in the sentence, while intents are the abstraction of the user's final intention. Intents can also be absent if the questions are merely conventional, but they are usually present, since there is almost always something that the user wants to know or execute [44].

For example, in the sentence "What are some good restaurants in Zürich?", Zürich could be an instance of the possible entity “City” and the intent would be “Find restaurants” [9] (a more detailed example will be given in 2.3).

Most of the bots today are of this type and can be based either on Artificial Intelligence or on a rule-based model. One possible example is a bot that is able to book tables at restaurants. Obviously, their specificity causes them to fail to answer more general questions that are outside of their domain [35].

2.2.2 Goal

Task-oriented

Task-oriented chatbots are designed to perform a specific task. The focus is therefore on a goal that the user wants to achieve, which compensates for the disadvantage of lacking general knowledge [22].

Generally, the course of the interaction with these types of dialogues are predetermined [35]. The general flow works as follows: the system interprets the user’s message and assigns an internal state, for which there is a reaction transformed in natural language. Language Understanding (LU) is processed by statistical models, but it should be pointed out that many systems still represent their states, intents and slot filling through manual features or hand-crafted rules. These kinds of dialogue systems are more time-consuming and expensive. For this reason, new algorithms that broaden the state space and use deep learning techniques have been an area of research in recent times. The latter help to improve the above mentioned aspects, because they learn to represent features in a high dimensionally distributed fashion [12].

Examples of these dialogue systems are hotel booking or customer support service. Implementations of this kind are difficult because of their application-specificity and lack of available training data [60].

Non-task-oriented

Non-task-oriented chatbots are not limited to a particular task, but focus instead on achieving a form of interaction closer to human interaction, which is continuous and does not have a predefined structure. Typical architectures behind this type of systems are either generative-based, for example Sequence-to-Sequence (Seq2Seq) models, or retrieval-based, choosing appropriate answers from a repository [22] (both are discussed in more detail in 2.2.5).

A fundamental requirement for these systems is a large volume of training data. They lack the ability to aggregate useful information in their feedback, since they are not able to support domain-specific tasks [60].

Two different types of non-task-oriented dialogues are information-oriented and conversation-oriented. Information-oriented chatbots return previously stored information to the user in a FAQ fashion using string matching, named entity recognition and response generation. Conversation-oriented bots are used to reach a continuous interaction, in which they try to respond to the latest input in the most appropriate way. They do this by using further techniques such as information retrieval and relevance detection, in addition to the ones already mentioned above [35].

To describe the most relevant techniques, string matching consists in the search of a specified string within another string. It can be either exact, having the goal of finding a perfect match of a specified string in a larger one, or approximate, searching for a pattern in the input string where a specific edit distance is at most a predefined number k [36]. Named entity recognition is a method that helps to classify named entities of a string into predefined categories. For example, it can help to identify people's names, locations or organizations in a text [29].

Creating an evaluation scheme for this model is impossible. Indeed, due to it not aiming to reach a specific goal, there are less evaluation metrics that can be taken into consideration (e.g. task completion rate). Moreover, each conversation is unique, it is dependent on the user's input and system's output and it has no defined end. One additional difficulty is the definition of a suitable database for the responses. Theoretically the structure can be given thinking of how the conversations with the future users are wanted to be. However, it depends on the chatbot design. There is not always a target of users and if the system is created for many different target groups, it is not clear what the best strategy for the design of the database is [64].

2.2.3 Service Provided

Interpersonal

Interpersonal chatbots are some sort of enablers that aim to collect information and give it as an output to the user. Therefore, they are more frequently service providers, such as booking and FAQ bots, rather than entertainment mediums, even though the latter should not be excluded as a possibility [35].

Intrapersonal

Intrapersonal chatbots have a more personal character and involve messaging application such as WhatsApp [1], Messenger [2] or Slack [3] in order to perform activities similar for example to calendar management. They act as a companion to the user, and are able to mimic human-to-human interactions [35].

Inter-agent

Finally, inter-agent chatbots are created in order to interact with other systems to perform a specific task. This is becoming relevant in the Internet of Things (IoT) domains. As chatbots become increasingly widespread, also inter-bot communication possibilities are becoming of interest. Such systems help improve the integration of different services thanks to communication. An example is the Alexa-Cortana integration [35].

2.2.4 Interaction Mode

Text-based

A text-based chatbot accepts inputs and gives outputs in the form of written text. They are suitable for desktop and mobile engagement, and they usually tend to be integrated into messaging services such as WhatsApp, Telegram or Slack, or into dedicated websites or applications [48]. For example, in 2017 PostFinance launched a web based chatbot to respond to customers' Frequently Asked Questions. This chatbot operates in German and it was the first chatbot launched by a Swiss financial institution [39].

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

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

[3] https://slack.com/

Communicating with text is better suited when a complex activity needs to be performed. When tasks require logic or include a confirmation step, text can help the user manage the interaction in a more structured and controlled way [65].

Voice-based

Voice-based chatbots, also commonly called voice assistants, are agents that can receive vocal input and respond via computerized voices. The most common examples are Apple's Siri, Amazon's Alexa and Microsoft's Cortana. Generally, they are integrated in smartphones or they are a separate device. A variety of activities can be performed. From merely answering questions to more detailed tasks such as playing music, home automation, or writing e-mails and to-do lists [20].

Voice-based chatbots are more suitable than text-based ones when multitasking activities are performed: they can help the user when it is moving or its hand or eyes are busy, so that it does not need to interrupt the current task [65].

2.2.5 Response Generation

Rule-based

The rule-based approach for developing chatbots is based on the specification of a set of rules or conditions [43]. Thanks to some identified distinctive variables a pre-structured answer is modified and outputted back to the user [40]. These distinctive variables can be specific words, phrases or actions [50].

More specifically, all the pre-structured answers are hard coded and linked to one or more keywords: when the system detects one of the specific keywords, a response is triggered according to it [16, 50]. The definition of the rules is crucial: without them the chatbot would not understand what the user is trying to ask. This is the main reason why chatbots of this kind are not really suitable for entertainment purposes [43]. Instead, this technique is often used for Q&A bots [40], where there is a limited number of foreseeable scenarios with a limited set of possible outcomes [35].

This method has its disadvantages. One of them is that the feedback given to the user is repetitive because previous answers are not generally stored in order for the system to learn from them. There is usually a set of predefined responses from which a suitable one is selected with only characteristic variables changing (e.g. name of a city). This has also the consequence that these systems tend to be non-human alike. It is unrealistic to think that a person would always answer in the same manner, as this would make the conversation quite uninteresting [40].

Retrieval-based

Retrieval-based approaches are more flexible models that use Application Programming Interfaces (APIs) for the querying and analysis of conversations. In this way it is possible for example to access information stored in a database and turn it into an output [19].

These approaches can be both based on a simple pattern-matching approach or a more compounded one such as Machine Learning. The idea behind it, is the application of heuristics to find the best-fitting answer within a set of predefined answers. As a consequence, there are no new insights produced by the algorithm. Instead, answers are always selected from a predefined pool of candidates [40]. The advantage is that the chatbot can get a fluent answer for a specific context [62] without making grammatical mistakes. On the other hand, the system is unable to carry forward contextual information from previous interactions, and it can therefore not refer to important information such as peoples' names or places [40]. The responses are also more predictable, which can have advantages and disadvantages. If the user wants to use a chatbot for buying a product, a retrieval-based model can be very helpful and can typically answer nearly all the queries e.g. about the product's specifications, its price, etc. This would not be the case for a conversation that goes outside of a specified domain. Hence, if the user would make grammar mistakes or try to chitchat with the system, the system would probably not understand [32]. Finally, the model requires a rich dataset to operate well [58].

It is common to use Artificial Neural Network (ANNs) for the implementation of these chatbots. This method takes advantage of neural networks to give a score to responses in order to choose the more suitable one amongst them [2].

Generative Model

The Generative approach allows to always produce new responses thanks to the exchanges of messages that have already taken place, and for this reason it is considered to be the smartest model [19]. The answers are generated with the help of a number of techniques, including for example machine translation [40] and deep learning [2]. The ability of referencing previous interactions makes the exchanges more similar to human interactions [40].

Nevertheless, it is difficult to build and train them in order to reach a good interaction because of the need for large datasets, which are not always easy to collect [19]. Furthermore, the answers generated by the model tend to be unforeseeable. For this reason, they are best suited for a non-task oriented application rather than a task-oriented one [32]. To give an example, the system could be biased during the training phases and could potentially include offensive comments that allude to racism or sexism.

Artificial Neural Networks are the technology on which Natural Language Processing (NLP) and Natural Language Understanding (NLU) solutions are typically based for the processing of input and generation of output [35]. Seq2Seq is an algorithm that is typically used for this purpose [2].

2.3 General Architecture

In this section, the general architecture of a conversational interface is presented based on the representation 2.2 which was introduced in [33] and then completed in [27].

When a request is received by the system, the Language Understanding (LU) component is in charge of interpreting the user's words and their meaning. Then, the Dialog Management component either formulates a response or if necessary asks users for further clarifications. The Dialog Management component also interacts with web services and knowledge sources to collect the required information to include in the response. The actual component that then constructs the response is called Response Generation (RG) [33]. This action of collecting information from external sources is labelled in [27] as Action Execution and Information Retrieval (AEIR) which interact with Data Sources (DS) in order to perform actions or collect useful information.

This representation is suitable for a system that recognises text input. If it were to also recognise voice input, it would lack the step of speech recognition before the Language Understanding step, and the step of text-to-speech synthesis after the Response Generation step for converting audio into words and words into audio back again.

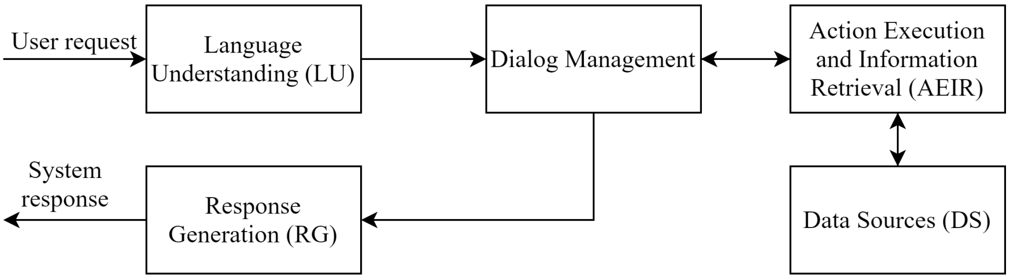


Figure 2.2: Own representation of the general architecture of a chatbot based on [33] and [27]

Kucherbaev et al. presented the functioning of the architecture by making use of an example of a conversation in which the user asks information about the weather. This example is summarized in table 2.1, and helps to better illustrate the architecture.

User: what is the weather?		
Language Understanding (LU)	Recognizes info	Intent: "check weather"
Dialog Management	Recognizes missing entity "location"	Entities: [location: ?]
Chatbot: For which city would you like the weather forecast?		
User: San Francisco.		
Language Understanding (LU)	Parses missing information	Entities: [location: "San Francisco"]
Action Execution and Information Retrieval (AEIR)	Performs requested action/retrieves information from its Data Sources (DS)	API response from openweathermap.org for San Francisco
Response Generation (RG)	Prepares response for the user	Response: "It is +23°C and sunny in San Francisco now."
Chatbot: It is +23°C and sunny in San Francisco now.		

Table 2.1: Own representation of an example that shows the functioning of the chatbot architecture based on [27]

As it can be noted in this example, the Dialog Management component helps identify the missing information needed to continue the conversation and to update the context (intent, entities). Other situations in which it takes action are when clarification by users should be processed (e.g. user asks about the weather for the day after) or follow-up questions should be formulated (e.g. the chatbot asks the user if he would like to see the weather forecast for the whole week).

2.4 Implementation

Chatbots can be created in different ways. In this section the most important methods for enabling their implementation are going to be mentioned.

A first option is to proceed by creating a chatbot with a programming language such as for example Python, Java, C++ or Ruby [3].These usually have Software Development Kit (SDK) libraries specifically created for developers to implement a conversational agent. Coding the chatbot allows developers to reach a higher degree of flexibility in terms of conversation building and integration with back-end tasks [23]. Two frameworks that fall into this category are Microsoft Bot Framework [4] and ChatterBot [5].

Another possibility is to use state-of-the-art platforms [3], which allow to build chatbots without the necessity to even write a single line of code. The conversational flow is often created with visual elements and maps. Chatbot components could even be drag and drop UI elements. Some examples are Chatfuel [6], ManyChat [7], DialogFlow [8] and Botsify [9] [23].

Obviously, there are also frameworks that have both a programming and user interface option. Botpress [10] and Rasa [11] are only two of them.

Finally, there exist chatbot scripting languages. With such languages, chatbots are created thanks to scripts containing conversational content and flow, which are fed into a rules engine or an interpreter program. The latter matches user utterances to scripts' templates in order to decide how to proceed with the conversation. A few examples of scripting languages are Artificial Intelligence Markup Language (AIML), ChatScript, and RiveScript. PandoraBots [12] is a web-based platform that can be used for creating AIML chatbots [23].

2.5 Applications: Companies

As already mentioned, chatbots are now spreading and finding applications in many different fields within our society. Not so many years ago, people communicated with each other mainly face to face or via phone. Now, not only it is possible to communicate through the internet, reducing the interaction-time consistently, but it is also possible to communicate with virtual agents that can take care of specific issues for you. Some personal assistants are mainly focused on individual users in order to obtain information and perform tasks (e.g. Siri, Cortana, Alexa, etc.). Others are built for enterprises, with the goal of providing their customers with support around products and services, whilst at the same time feeding useful information back to the company [33].

This is how the labour market is changing today: repetitive and well-defined tasks can be automated, leading to increased efficiency and savings. Chatbots can help many different kinds of businesses in improving both business internal processes, and services provided to their customers [48].

Customers may be pushed to use a chatbot out of curiosity, but this is not enough. The purpose of companies is to create an additional value to the customer that improves the current interaction and convinces them to keep using the new technology beyond the initial spark that aroused their interest [65].

Even if this seems pretty obvious theoretically, there are always many aspects to consider if a company decides to introduce an agent in order to help its business. Is the chatbot really helping the company in terms of growth, image and customer satisfaction? What if, for example, the maintenance costs of the agent are way higher than the additional profits it generates?

[4] https://dev.botframework.com/

[5] https://chatterbot.readthedocs.io/

[6] https://chatfuel.com/

[7] https://manychat.com/

[8] https://cloud.google.com/dialogflow/docs

[9] https://botsify.com/

[10] https://botpress.com/

[11] https://rasa.com/

[12] https://home.pandorabots.com/

This consideration is not the main focus of this thesis, but it deserves mention as it is a central question nowadays. For this reason, an example in the insurance field is presented in subsection 2.5.1, as it is a good way to reflect on how many different areas chatbots can be applied in. This is good introductory food for thought to be able to subsequently present a concrete use case in chapter 3.

2.5.1 Example: Insurances

Even a complex sector such as the insurance one slowly changed and started to evolve with the help of new digital possibilities. Nowadays, a customer-centric approach helps businesses conquer a big advantage in the market. The trend is towards never being disconnected and always being available, offering a 24/7 service [49]. Since the activities of insurance companies include many different kinds of exchanges between the company and the customers, chatbots can help insurance companies in this context, as they can be used to automate structured tasks [24].

Chatbots can help insurances in many different areas. Singh et al. listed no less than ten different categories in which they can find applications. Registering a claim, showing the policy's status, suggesting a plan or a policy and finding an advisor are just some examples. These types of queries are usually handled by people working in call center, and it was estimated that about 70% of them could possibly be handled without any human involvment. Given the costs generated by call centres, chatbots could even take half of the calls and save a considerable amount of money. Moreover, chatbots do not have a limit on the number of requests they can deal with at any one time, allowing to considerably reduce the waiting time for customers. A customer who browses a company's website seeking information usually takes 5 to 10 minutes and a call center agent has an average wait time of 3 minutes. Instead, chatbots can efficiently give back information in real time [49].

Analytics are also important in this context and should be mentioned. Chatbots could help define patterns for each customer in order to offer them a more customized experience and potentially suggest new suitable products [24].

A concrete example is the chatbot of the company Insurify. Launched in 2017, it has now become the first insurance comparison platform in the U.S., which is integrated with Facebook Messenger and is able to provide car insurance quotes, recommendations and advice. It works in the same way as calling an agent and the interaction only takes about two minutes. Other than comparing insurance policies, the chatbot allows the user to subscribe to rate alerts, which notifies him when the prices drop or when a more convenient offer is available with another company [26].

2.6 Opportunities and Challenges

As discussed in section 2.2, there are many different types of classification criteria for chatbots that can be combined in many individual ways. The consequence of that is that the advantages or disadvantages of a chatbot are partially dependent on the specific solution that is chosen. For this reason, the opportunity and challenges of chatbots are going to be discussed in general terms and do not necessarily apply to all the possible instances.

Generally speaking, interacting with a chatbot is not so different to most people's daily exchanges. Most of the time, it involves the same devices and applications that people are already familiar to interacting with, taking advantage of informal communications. Moreover, since nowadays technology is everywhere, they are not limited to unique fields. We can find examples of chatbots in Education, Healthcare, Marketing, Entertainment, Supporting Systems, Cultural Heritage and many others [3]. Therefore, getting started is pretty straightforward: it is sufficient to use our computer or phone and open our WhatsApp, Telegram or Facebook Messenger applications [61]. These are used instead of calls, e-mails or websites. The only drawback is that some customers may not be used to this change and may therefore need some time to adapt and to be convinced that they are getting a real benefit from it [66].

Chatbots change the normal flow of interaction and are a big technological advancement. It should be considered that they leverage one-to-one communication which, compared to traditional channels where humans are still involved, allows an interaction that crosses the time limits of a normal business. The fact that the virtual agent can be reached 24 hours a day and 7 days a week is a big advantage for better meeting customers' needs, both on a local and international scale [66].

A 2020 study by Abdellatif et al. examined the topics that chatbot developers are most interested in and their challenges with the help of Q&A on Stack Overflow. The most discussed topics included development, integration and NLU and the ones that were found to be more helpful in building and integration. The results showed that developers struggle particularly with the training of their chatbots' systems [1], since it requires a large amount of data for the system to produce adequate answers. Furthermore, the data provided could also be inappropriate. For example, it could be too broad or specific and therefore it would not guarantee the normal flow of a conversation [30].

In a 2017 study conducted by Zamora, participants indicated their main expectations around chatbots. Four main categories could be identified: a virtual agent should be high performing (e.g. speed, efficiency and reliability), smart (e.g. knowledge, accuracy, foreseeing skills), seamless (e.g. ease of use, flexibility) and personable (e.g. ability to understand the person, likeability) [65].

Another study by Folstad investigated the main reasons that motivate the use of chatbots. Results indicated that the main reason for using a chatbot is productivity: 68% participants specified their ease of use, speed and convenience as motivations. 41% of the participants also indicated that these agents can really be of support for obtaining help and information. Other aspects that motivate the use of chatbots were entertainment (20%), social and relational purposes (12%) and their novelty (10%) [8]. This suggests that efficiency is really important for chatbots' users.

Speaking of the performance of a chatbot, the user's perception is generally relative to his own experience. This has as the consequence that the expectations in terms of speed and efficiency are dependent on the user and his own standards. For example, many people still prefer the use of a standard search engine for what concerns information seeking or the completion of complex tasks [65].

A point in favour of chatbots is the absence of judgment that the user perceives in interacting with them. Many users may be embarrassed to ask certain questions, for this reason they may feel more confident to ask them to a robot rather than to a real person [65].

Semantics is important for the dialog system for understanding the message of the user on one hand, and for detecting the user's emotion and needs during the interaction on the other. Technically, this matter can be reflected in NLU techniques such as entity recognition and linking, domain/topic/intent and user sentiment/emotion/opinion detection and knowledge/commonsense reasoning. Another challenging aspect is consistency: the user should have a unique vision about the system and its "personality". Only in this way, can the system be convincing and gain the user's trust. Three important dimensions are personality, style, and contextual consistency. Interactiveness is the key for creating entertainment and satisfaction. The system should be able to understand the user's social needs, among which are emotional affection and social belonging, and hence utilize this information to its advantage for adapting the response-behaviour [21].

In [27] it is shown that chatbots still have some limitations in different areas of their architecture such as Language Understanding (LU) or Response Generation (RG). According to this paper, an intuitive solution, which is currently used in different types of systems, would be to have a human-in-the-loop, who has access to the web and is an expert in the topic of conversation. This concept is called human-aided bots. Nevertheless, the degree to which humans are involved should be better investigated to find the right balance depending on the domain. In the future, the participation of a third party is expected to decrease, due to the improvement of automated systems in performance and adaptability.

Another opportunity presented by chatbots is the possibility to collect personal data about the customers. For example, if the chatbot is integrated in Facebook Messenger, the information in users' profiles can be directly extracted. Another possibility is that the chatbot collects information within the conversation flow with the user. This information is then an opportunity for the companies to improve their understanding of their customers, and potentially use them for personalized marketing [66]. This is often in contrast with the challenges of data privacy and data protection. Users fear that their data will be disseminated or mishandled. This is especially the case when personal information, such as information on personal finances or social media, are handled [65]. It must be taken into account that these agents are created by companies that have their own interests at heart, and therefore it is not surprising to encounter scepticism on the user's side. Since the chatbot is collecting personal data, the user should be informed on how the data that he is providing is used and should agree to it. Also to be considered is that often chatbots rely on third-party platforms (Facebook, Telegram, etc.), so developers need to be careful [66]. This shows all the more reasons why it is important to gain the trust of the user when sensible data is involved [65].

All the identified opportunities and challenges are summarized in table 2.2.

Opportunities	**Challenges**
Generic	*Generic*
• Similarity with the usual means of communication [61] • Possible application in many different fields [3] • Suitability for easily accessing information [8] • Absence of judgment in the interaction [65] • Chance for knowing the customer better [66] • 24/7 availability [66]	• Possibility of long adaptation times [66] • Personal perception about performance [65] • Keeping the system interactive [21] • Need to understand user's social needs [21] • Finding the right balance of human involvement [27] • Data Privacy [65]
Technical	*Technical*
• Productivity (speed, ease of use, convenience) [8]	• Difficulty of the training [1] • High expectations on performance (speed, ease of use, convenience) [8, 65] • Applying NLU techniques considering the importance of semantics [21] • Limitations of the NLU techniques themselves [27] • Having a consistent system [21]

Table 2.2: Representation of identified opportunities and challenges with corresponding literature

Chapter 3

Use Case

This chapter is the application part of this thesis. Specific contextual scenarios for which a chatbot can be useful have been defined through a collaboration with employees from the Department of Informatics of the University of Zurich (IfI). After a short presentation, it will be explained how an interview conducted with them took place and allowed the definition of the scope of the investigation. Then the most fundamental part will be dealt with, i.e. the requirements that were defined as a result of the interview and the resulting design decisions will be listed. Finally, a possible functioning of the chatbot, possible design solutions and some mockups will be presented.

3.1 Presentation

The UZH Department of Informatics (IfI) is part of the faculty of business, economics and informatics and it offers Bachelor, Master and Doctorate study programs. The IfI studies administrative assistant and academic coordinator deal with departmental issues on a daily basis. Their competence focuses particularly on the Bachelor and Master thesis procedure and other study-related topics such as the management of a mandatory internship that the students need to do.

3.2 Interview

An interview with IfI's academic coordinator and administrative assistant was conducted with the aim of formulating a more concrete proposal for a chatbot that supports them for some of their activities. In this regard, there were a lot of questions that needed to be answered. The most relevant ones are going to be presented and discussed in this section. It was known that the department had to face many questions every day, but it was necessary to define who are the stakeholders that generate most of its work.

***Q**: From which stakeholder do most of the questions come from?*
***A**: More than half of my interactions are with students. Mostly students that are already enrolled [...].*

In the department, there are more than a thousand students. IfI's employees do not know each of them personally, as opposed to the professors who do. For this reason, the focus of the use case was put on the interaction with the students. The next step was understanding what activities carried out by the department could be suitable for a chatbot. In order to do so, it was important to have a clearer idea of what their activities are. A good starting point was to analyse some of the Frequently Asked Questions.

***Q**: [...] Can you explain what are the Frequently Asked Questions that you receive?*
***A**: Frequently Asked Questions are typically regarding the thesis. How do I have to hand in my thesis? Where do I find that? Very often I get asked what the admission requirements are for prospective students. On a daily basis I point them towards the dean's office or to the admission website. This is something that comes in daily. Or questions regarding fact sheets or registration forms in response to which I just provide a link. Or how can I get my ECTS credit? Then I forward them the link to the dean's office website where everything is explained. Can I use a language course for my study? How can I do that? And then there is another link. So, often I just reply by sending a link to the website or a link to the Course Catalogue. Can I use that course for that? You find it in the Course Catalogue. Where is the room? You find it in the Course Catalogue.*

It can be noticed that the dean's office is mentioned above. The dean's office takes care of the most general activities. An explanation of what it does was very relevant in order to understand IfI's real competences. This was clarified during the interview:

In the past all the processes were managed here. No students ever had to go to the dean's office. Then we realised that many processes were the same. So we started to handle all the processes for IfI and for oec students in the dean's office and we only kept the informatics-related processes. That's why we don't have that huge number of general questions from a big number of students. Then we reduced our website to that one site, because all relevant information is on the dean's office website. We have just a few specific things and we try to put them all in one website. That's why we try to forward to the dean's office. Whenever I think that question is not for me, I just send it to them, so the students learn that the dean's office is responsible for that. [...]

Not all the students figured out this separation, and many questions came to the department for this reason. Looking at the IfI website, it can be understood why there is confusion about who to contact in case of questions. In the home page, three different paths can be followed to obtain three different contact information. If the "Bachelor" or "Master" button is clicked the personal contact of the coordinator is indicated. If it is decided to click the "Contact" quick link, two other different emails are shown instead: info@ifi.uzh.ch for general questions about the department and studies@ifi.uzh.ch for questions about the informatic study and the internship. Finally, clicking on the "Student Advisory Services" quick link, the coordinator, the dean's office and other four email addresses are shown, but this time with the competences and support areas clearly indicated for each of them.

Suitable activities for a chatbot are usually simple and repetitive. For this reason, it was necessary to further investigate which activities qualify as such.

***Q**: What are the most repetitive activities that you perform?*
***A**: This is very hard to tell because there is a certain number of emails that only require sending a link and pointing them to where they can find the information. And then still, as we are the only contact persons for IfI-related questions, they are often individual. So, they need personal attention, because they think they have a special case, or they have a special question or a special situation.*

Activities for which students need personal attention are usually individual and difficult to standardise. Therefore, the focus was put on the kinds of activities for which the department only had to respond by providing a link to a website. Obviously, the goal of the chatbot is not to cover all the activities performed by the department, but rather to get rid of or reduce the frequency of the most tedious ones.

3.3 Scope Definition

To sum up, the coordinator receives several emails about issues that are not of her competence. The ones for which usually only a link is forwarded in response are certainly a good target for a chatbot. They are simple to deal with, they are repetitive, and they waste the coordinator's time to engage with the tasks that she is responsible for. In the interview the coordinator pointed out that the question-answer cycle for these kinds of matters is two to three days, since they are not her primary concern. A chatbot is available 24/7 and is able to handle some of these situations in a few minutes. These activities can be easily organized in a FAQ-fashion, that can also easily be extended with further topics.

A first suitable task identified for the chatbot for supporting the coordinator is indicating whom to contact according to the type of problem presented by a student.

Another identified activity for the FAQ category is the graduation process. All the information about it can be found on www.oec.uzh.ch under the studies information. Here the students can find important details such as where to find the graduation form and the relevant deadlines.

Regarding the activities for which the IfI is directly responsible, the Bachelor and Master thesis procedures have repeatedly been mentioned as areas of particular interest for students.

Therefore, the thesis procedure is the last activity identified as suitable for the chatbot's support. In order to understand how it works, students have a fact sheet at their disposal on the IfI's website pages called "While Studying", both for Bachelor and Master. First of all, they need to find a professor and a topic for their thesis. Then, they need to complete a registration form containing all the relevant information, including e.g. the start date of the thesis. This will be approved only if the student has already completed the assessment level (first year of studies) and the mandatory internship. When the coordinator receives the form, she needs to check these requirements on an internal Enterprise Resource Planning Software (SAP). She then enters the start and end dates of the thesis on an Excel file. This is a standalone process that the chatbot will be in charge of. It would have to help with the registration and the lookup for the submission of the thesis. Students also often show interest about particular topics, that the coordinator has to explain over and over again. Examples are writing the thesis with a professor from ETH rather than UZH, writing it with an external company, how to fill out the form thesis form, relevant deadlines and the submission procedure once the thesis is completed.

The most suitable areas of interest for the chatbot are presented below in 3.1.

3.4 Requirements

Thanks to the interview, a current situation analysis, and the scope definition, it is possible to define what requirements the chatbot use case must fulfil. In this section they are being discussed for the categories user, functional and non-functional requirements. A precise requirements catalogue can be found in appendix A. For a better and precise visualization of how some parts of the interview were turned into requirements instead, a visualization is provided in appendix B.

3.4.1 User Requirements

Thinking of the end user, a first important general requirement is to always have the possibility to get the right information depending on whether you are a Bachelor student or a Master student (UR1).

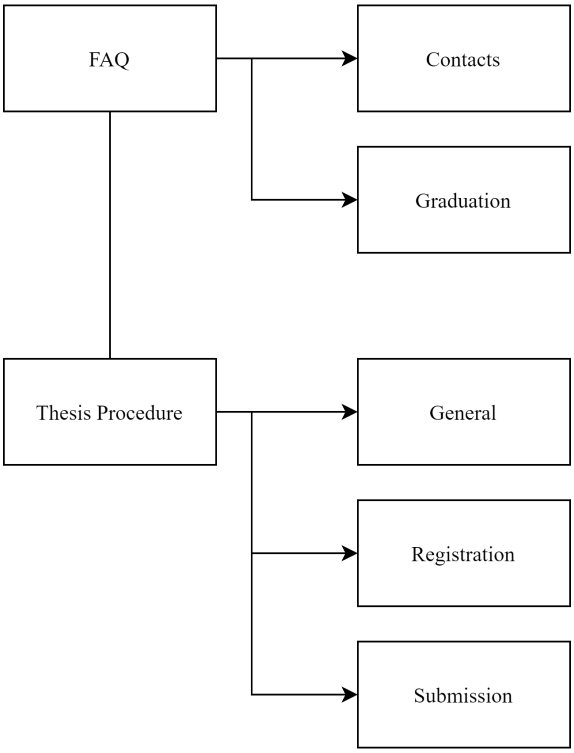


Figure 3.1: Areas covered by the use case chatbot

Frequently Asked Question (FAQ)

The goal of the students for the contact use case is simple. They want to know who is the right person/department to contact depending on their question (UR2). For the graduation use case instead, they want to know what they should do in order to obtain the degree (UR3), this includes the possibility to download the appropriate graduation form (UR4) and see their degree deadlines (UR5).

Thesis Procedure

When it comes to the thesis procedure, students want to get information about what the thesis is (UR6), how many credits it is worth (UR7), how long it takes (UR8) and what their requirements are in order to be able to get started (UR9). They also want to know if they can write the thesis with support from external contributors such as ETH professors or independent companies, and how it is organized (UR10). Additionally, students want to have the possibility to directly download the registration form (UR11) and have clarification about where they need to sign (UR12). The submission is also important, the submission procedure should be explained (UR13) and students should always have the possibility to check their deadline via authentication (UR14).

3.4.2 Functional Requirements

The chatbot should accept text and button inputs (FR1) and be able to respond both with text and files (FR2). It should accept either English, German or one of the two (FR3). The final result should work on different OS (FR4) and should function 24/7 without unnecessary interruptions (FR5). The data for the chatbot should be retrieved from different kinds of websites (FR6) but can also be manually inserted and modified (FR7). If the data is retrieved from a particular webpage, the chatbot should indicate where the information was taken from (FR8). When the user needs to make a decision, the system should suggest possible input to the user (FR9). Finally, it emerged in the interview that IfI's social media platforms (Facebook and Instagram) should not be considered as a means to integrate the chatbot (FR10).

Frequently Asked Question (FAQ)

For the contact part of the use case, the system should be able to extract the contact information (FR11) and the area of concern of each of them (FR12) from the Advisory Services page on the IfI website. Unfortunately, this page does not contain all the necessary information: both contact information (FR13) and responsibilities (FR14) of the dean's office should be taken from www.oec.uzh.ch. For the graduation part instead, the link of the graduation form file should be retrieved from www.oec.uzh.ch/en/studies/general/regulations.html (FR15). Then, the degree deadlines should be retrieved from www.oec.uzh.ch/en/studies/general/graduation/degree-deadlines.html (FR16).

Thesis Procedure

All the information about the thesis procedure should be extracted from the fact sheet PDF, that is linked in the "While Studying" pages on the IfI website (FR17). In regard to the thesis procedure, the link to the registration form should be extracted from the IfI website (FR18). Finally, it should return the deadline for the ongoing thesis for each individual student only if an authentication has taken place (FR19).

3.4.3 Non-functional Requirements

Generally speaking, the system should be intuitive (NFR1), easy to use (NFR2) and it should have short response times (NFR3). When it comes to the deadline control of the thesis procedure, privacy should be considered (NFR4).

3.5 Design

This section presents the complete design of the IfI chatbot. 3.5.1 formalises all the design decisions that have been made. In order to see the specific decisions that were based on the interview's findings, see appendix B. Based on these choices, subsection 3.5.2 shows how the internal functioning of the chosen solution can look like. In 3.5.3 some practical solutions for the implementation of the chatbot are proposed. Finally, 3.5.4 illustrates two examples of possible interactions with the chatbot through Mockups.

3.5.1 Decisions

General Chatbot Classification

In regard to the chatbot direction, some basic decisions have to be made before discussing the design approach more concretely. The classification of chatbots presented in section 2.2 is a good starting point for an initial idea of the chatbot direction. All the choices that will be made will be justified with the theory discussed in that section.

Open domain chatbots have no limitation of domain for the interaction [35], they can talk about almost every topic and therefore the answers are usually quite generic [4]. Closed domain chatbots are suitable for a specific knowledge domain [35] and they work thanks to the classification of the most meaningful keywords into entities and intents which are recognized and trigger the suitable response [44]. The IfI chatbot belongs to the latter category, since it has a predefined domain, namely the university one. Answering general questions such as "What is the weather in Zürich?" is not the reason for which the chatbot is designed and it should therefore not be part of its scope.

The goal of the chatbot can be distinguished between task-oriented and non-task-oriented [35]. The first category of chatbot deals with specific tasks with a specific goal, such as a hotel booking agent [60]. Non-task-oriented chatbots are not focused on a specific task but rather on a form of interaction similar to the human-one [22]. They can be information- or conversation-oriented: the first one usually has a FAQ structure and should return information, while the second is more about continuous interaction, for the purpose of making conversation [35]. For this particular use case, the goal for the students is mainly to receive information for clarifying their doubts about specific topics. This would suggest that an information-oriented chatbot is most suitable. However, certain parts of the chatbot also need to trigger specific actions, such as downloading the thesis registration form or checking the submission's deadline. It can therefore be said that the IfI chatbot is a mix between a task-oriented chatbot and an information-oriented chatbot.

A chatbot provide three different kind of services. The interpersonal service has the goal of collecting information and give it as an output to the user, such as FAQ bots. The intrapersonal service is of more personal character and help the user with personal tasks mimicking the human-to-human interaction. The inter-agent service refers to the interaction between two systems and is relevant in the IoT domains [35]. The service that the IfI chatbot should provide is the interpersonal one, since the interaction happens between the system and the user with the goal of obtaining information on certain topics.

Chatbots can interact either via text or voice. A voice-based chatbot is usually more suitable for multi-tasking activities [65], while text-based chatbots are more indicated for complex activities that require keeping track of the conversation structure. Students would probably use the chatbot because they have specific question and only focus on that. The interaction mode should therefore mainly be text-based, but also buttons and files such as the registration form for the thesis and the degree should be incorporated (FR1-2).

Response generation modes can be categorized into rule-based, retrieval-based and generative models [19]. Rule-based approaches are defined by some if-else conditional logic. The conditions are then assigned to a set of rules with predefined patterns for each of them [51]. Retrieval-based approaches are similar, with the additional feature of being able to use machine learning mechanisms to train the chatbot. The training allows the chatbot to be confronted with examples, so that it can observe patterns on the data and predict the cases that would come up more often. Both approaches do not produce new insights, but simply select the output from a pool of predefined responses [40]. Patterns should be programmed manually, and it can be tedious if a lot of scenarios need to be distinguished. Generative models are "smarter", because they can learn from past responses. However, they are rarely used because they require the implementation of complex algorithms and a very large dataset [51]. Considering the scope of the IfI chatbot, a rule-based approach would be appropriate and should perform well enough. The generative approach option can be discarded, as the scope is not particularly broad and can be reduced to a set of well defined scenarios. A retrieval-based approach is also less suitable as it would require a rich dataset in order to operate well [58].

In table 3.1 the final chatbot classification for the particular use case is presented.

Knowledge Domain	Goal	Service Provided	Interaction Mode	Response Generation
Closed	Information-oriented and Task-oriented	Text, Button and File	Interpersonal	Rule-based

Table 3.1: Chatbot classification for the IfI use case

Text and File Responses

The FR2 specifies that the system must be able to respond with both text and files. Responding with text would not be a problem as it is the standard way of interaction for a simple chatbot. The files that the chatbot needs to be able to provide are the application form for the Bachelor and Master thesis and the graduation form, which are in PDF or Word format. On the website, there are links that allow the students to download all the files on their computer. The simplest way for the chatbot to provide these files is to extract and save these links. In this way the responses would be text-based, but the files could be easily accessed from them. Furthermore, if the files are modified, the chatbot can be updated by simply extracting the link again (see 3.5.1).

Language

In FR3 it is specified that the chatbot solution should interact in German, English or both. Indeed, it was mentioned in the interview that the interactions with the IfI student office are conducted half in German and half in English. Reasoning from a practical point of view, most commonly, students who speak German are also able to speak English, while the converse is not usually true. Furthermore, from a technical point of view there are certainly more possibilities for implementation in English than in German. Finally, choosing only one language for the chatbot reduces its complexity. For this reason, it was decided that the chatbot should only be implemented in English.

24/7 Operation and OS

In order to create advantages for students compared to a human-to-human interaction with the staff at the student office, the chatbot must be accessible at any time. This dictates that it must be hosted on a server that runs 24 hours a day everyday. More specific decisions can not be made, as these will depend on the specific implementation mode that is chosen for the chatbot. Many frameworks make their own servers available, whereas implementing the chatbot with a programming language often requires a personal server available. The same uncertainty also applies to FR4. It has been established that the chatbot should be able to work on a range of different Operating Systems. However, it can not be known whether it will work or not until the system is actually implemented. The idea is still that it should work at least on Windows, macOS and Linux. This will require efforts in terms of testing.

Data Extraction

It was already pointed out that the chatbot should be able to provide the information that is found on the IfI and oec websites. In the FR6, FR8 and FR11-19 requirements it is specified that the needed information should somehow be extracted from these websites. Some possible methods of extractions are going to be discussed in order to find the most appropriate one for this use case.

The first option would be to browse the IfI and oec websites, manually extract the necessary data, and write it in a JSON, CSV or XML file. There are several reasons why this is not the ideal method. It would require time and effort to set up initially, and most importantly it would not allow flexibility in the sense that any changes to the information on the website would require a manual update to the file. This would require a constant supervision of the website to ensure the correctness of the data.

A good idea would be to use a REST API to extract data from the website. Unfortunately, the University of Zurich is not yet providing an application that could help in that sense.

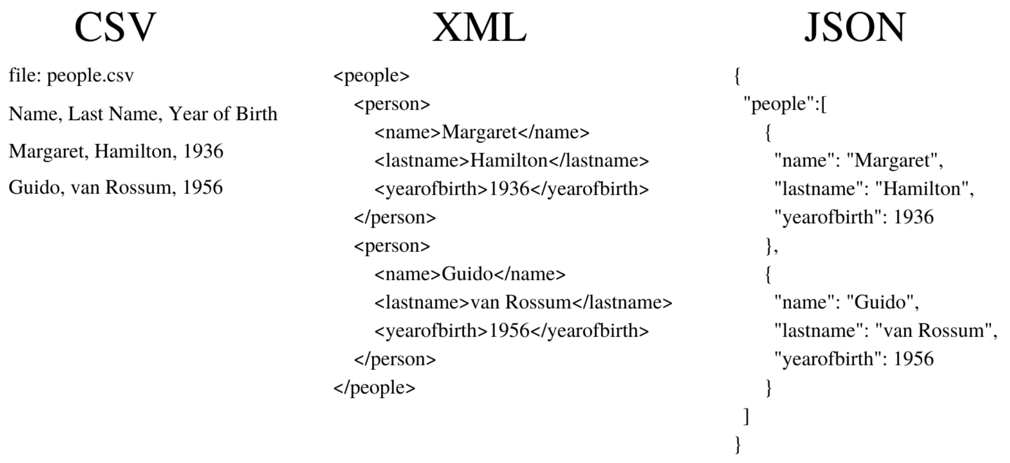
For these reasons, the best method for extracting all the needed data for creating a chatbot is web scraping. The web scraping technique helps to extract unstructured data from websites and give it a structure. The data can be structured in many ways: as databases, CSV, JSON or XML files [45]. Each piece of information that is extracted must be saved as a data record, alongside the specification of which particular web page it refers to. This allows the FR8 requirement to be met.

Comma Separated Values (CSV) is a way of storing data in plain text files. A CSV file is divided in lines and each line represents a new record. A record can have many values, which are separated by a comma. This format is very convenient because it can be recognized by a large number of applications [17].

Extensible Markup Language (XML) is a way of storing and transporting data, and its purpose is to represent data in a structured way. It is similar to HTML, with the difference that XML is suitable for carrying data and HTML suitable for displaying it. It is both human- and machine-readable. Its documents have a tree structure: they start from a root element which can have several children, which can themeselves have children, and so on. It operates with tags that can be defined from the author of the document [56].

JavaScript Object Notation (JSON) is a text format and is also a method for storing and transporting data. In its syntax, data is represented through name/value pairs and separated by commas. Curly brackets contain objects, and square brackets contain arrays. The main difference between XML and JSON is that XML needs an XML parser to be parsed, while for JSON a standard JavaScript function is sufficient [55].

In 3.2 an example that illustrates the difference between these three formats is presented, while in 3.3 a more detailed view of Web Scraping is given.



CSV

```
file: people.csv
Name, Last Name, Year of Birth
Margaret, Hamilton, 1936
Guido, van Rossum, 1956
```

XML

```
<people>
  <person>
    <name>Margaret</name>
    <lastname>Hamilton</lastname>
    <yearofbirth>1936</yearofbirth>
  </person>
  <person>
    <name>Guido</name>
    <lastname>van Rossum</lastname>
    <yearofbirth>1956</yearofbirth>
  </person>
</people>
```

JSON

```
{
 "people":[
   {
    "name": "Margaret",
    "lastname": "Hamilton",
    "yearofbirth": 1936
   },
   {
    "name": "Guido",
    "lastname": "van Rossum",
    "yearofbirth": 1956
   }
 ]
}
```

Figure 3.2: Difference between CSV, XML and JSON file formats

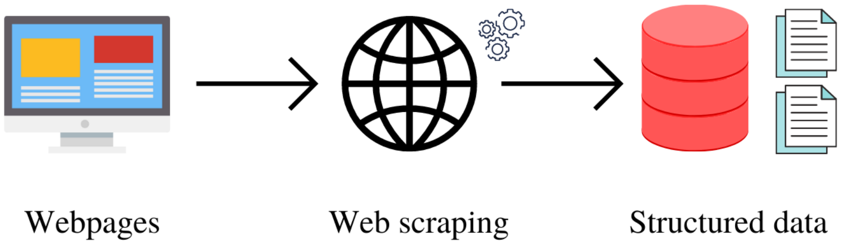


Figure 3.3: Web Scraping Architecture

There are modules that help programmers to perform Web Scraping. In Python, this can be done for example, using regular expressions, BeautifulSoup or lxml. Regular expressions allow the most time-efficient scraping, but expressions can quickly become difficult to construct and unreadable. BeautifoulSoup[1] is a library for extracting data out of HTML and XML files, which provides a useful interface for navigating the content. Lxml is a wrapper for the libxml2 library written in C for the parsing of XML. It is faster than Beautiful Soup, but its installation is hard to complete on some computers [28]. In 3.1 it is shown how this is possible with BeautifulSoup. In the example, a Python function is able to extract the link of a specific file if some defined keywords are part of the file title on the website.

```
from bs4 import BeautifulSoup
from urllib.request import urlopen

def find_file(link, conditions):
   file_link = ""
   with urlopen(link) as response:
      soup = BeautifulSoup(response, 'html.parser')
      root = "https://www.ifi.uzh.ch"

      for link in soup.find_all('a'):
         # If all the conditions are in the link's title then is the right one
         if all(condition in link.text.lower() for condition in conditions):
            href = link.get('href')
            file_link = root + href

      if not file_link:
         file_link = "Not found."

      return file_link
```

[1] https://beautiful-soup-4.readthedocs.io/en/latest/

```
#Find the registration form for the Bachelor thesis
find_file(’https://www.ifi.uzh.ch/en/studies/bsc-info.html’,
        ["registration", "bachelor", "thesis", "docx"])

Output: ’https://www.ifi.uzh.ch/dam/jcr:caa92451-b9c5-4f4f-bcb6-83263a15446b/
AnmeldungSchriftlicheArbeitenRVO16_en.docx’


#Find the fact sheet for the Bachelor thesis
find_file(’https://www.ifi.uzh.ch/en/studies/bsc-info.html’,
        ["fact sheet", "bachelor" "pdf"])


Output: ’https://www.ifi.uzh.ch/dam/jcr:8c1bb436-d2c4-4425-9264-167ed9b27c8a/
Merkblatt-BachelorarbeitRVO16_en.pdf
```

Listing 3.1: Function that finds the links of the files in the While Studying page

Hints/Buttons

In respect to the FR9, NFR1 and NFR2 requirements, one way for meeting these would be for the chatbot to always present the user with a range of possible choices. This can be done by simply writing hints when the chatbot asks a question to the user, or by having the user select an input using buttons. This simplifies the process for both the user and the chatbot, which already knows the topic on which the questions will be asked. Some frameworks already offer the option to create buttons to select certain criteria. On the other hand, if the chatbot is implemented with a programming language, more effort in terms of User Interface development would be required.

Integration

Nowadays chatbots can be integrated into many different applications. In FR10 it was pointed out that Facebook and Instagram should not be considered as possible integration channels. This is due to the fact that the IfI staff does not use them to share study-relevant information. Since the information that the chatbot shares is part of a university environment, which is professional, any application that people generally classify as being for personal use is not to be considered to be suitable. There are two interesting options to consider. The first one would be to integrate the chatbot directly into the IfI website or in a separate website linked from it. This has the advantage that is immediately noticed by the students when they are browsing for information. The second one would be to integrate the chatbot into Microsoft Teams. This approach would be suitable for every student, since as a consequence of the Coronavirus Pandemic they were all provided with a personal Microsoft Teams account. This chatbot could also be linked on the IfI webpage.

Authentication for Checking Submission’s Deadline

In order to satisfy the requirements UR14, FR19 and NFR4, the students’ privacy must be taken into consideration. The coordinator mentioned that the deadlines of ongoing theses are non-public information: they are written in an Excel file and are only accessible to the people that play a role in the thesis. Therefore, it is necessary to have a sort of authentication in order to identify each student. One way to do this is to interface with the UZH authentication method from the chatbot. Other external UZH services already do. This should not be a problem once that UZH grants access to the information of their identity management system [2], for instance thanks to the OAuth authentication protocol allowing the chatbot to interface with the university system. This gives access to the authentication system via a token, so that the access data is not directly exchanged between the two systems [52].

3.5.2 Functioning

Once the relevant data is extracted from the website via web scraping, it is possible to construct the structure of our chatbot. The methods can vary depending on the technology that is wanted to use to create the chatbot. In the Data Extraction subsection, it has been explained that the relevant information for the use case chatbot can be stored in different file formats such as CSV, XML or JSON files. However, the responses still need to be formulated and organised in a convenient structure.

Since a rule-based chatbot has been chosen, the answers set for each possible question must be firstly defined and then manually created (FR7). The information extracted through scraping can then serve as variables for the construction of the answers. The use of these variables makes it possible to be flexible in updating the answers if they are changed on the website. Nevertheless, care must always be taken, as trivial changes to the IfI and oec websites are enough to cause scraping to produce errors. The dataset must therefore be thoroughly cleaned at a later stage.

Responses must be defined depending on the implementation chosen for your chatbot. Implementing a chatbot with a programming language such as Python has the advantage of being able to insert the variables directly into a list/dictionary of questions and answers. Furthermore, the variables can be imported and converted in many different file formats. In addition, Python libraries such as Natural Language Toolkit (NLTK) [3] can be imported, allowing for instance the use of lexical databases such as WordNet.

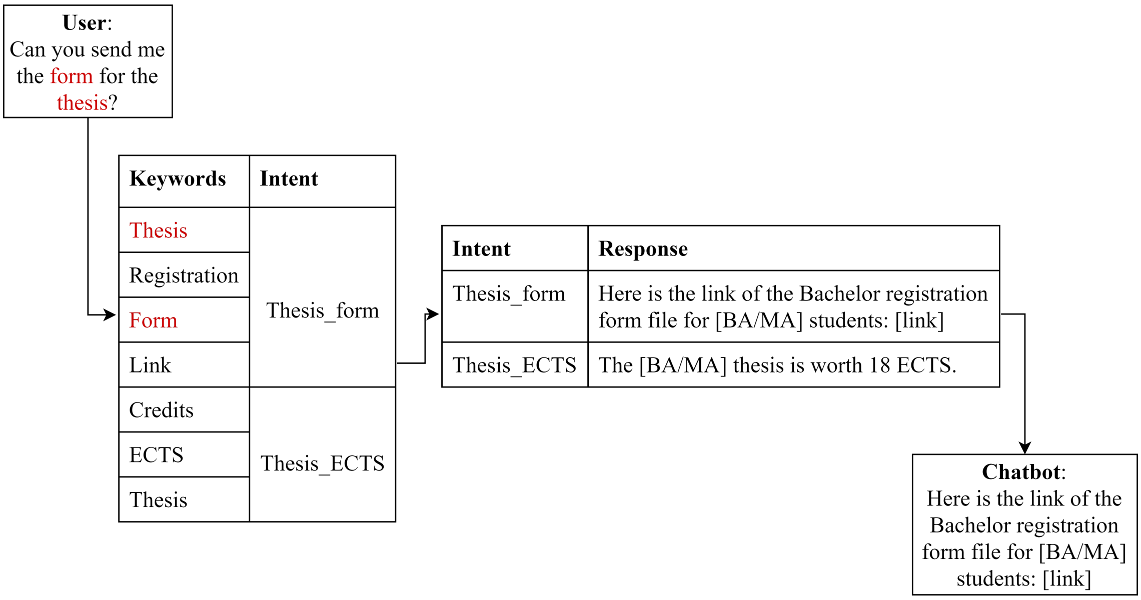
If, on the other hand, the use of a specific Framework is chosen, the capabilities of the system will depend on the capabilities offered by the framework itself. Some may offer the ability to organise web scraping directly from the program, others may allow you to import data from a file and others both or neither. For example, Google's DialogFlow, allows to directly import the intents from a CSV, where possible questions and responses are indicated. It is enough to indicate the columns intent id, intent, query and response [5]. Each framework is a different case and its required methods and formats may change.

Considering the relevant design decisions made in 3.5, a possible internal functioning of a rule-based chatbot is presented in 3.4. In order to access the responses, the user should trigger some keywords that are collected under specific intents.

But what happens if the bot is not able to recognize any keyword in the user message? In this case the chatbot should respond saying that he did not understand and suggest possible scenarios of why this is the case. The two main reasons for which the chatbot fails in its response are the presence of grammatical errors and the requests for information that are out of context. A text explaining what the scope of the chatbot is must be presented to the user. If instead there are an equal number of keywords triggered for each intent, the chatbot should ask for clarification: some sentences asking the user what particular topic he refers to and what options are available should be ready. Some variables that refer to the intent name should therefore also be stored. For example if a user sentence only contains the word "Thesis", the chatbot would reply "Are you asking about the [Thesis form] or the [Thesis ECTS]?". Thesis form and Thesis ECTS would be the variables for this case.

[2] https://identity.uzh.ch/itim/ui/Login.jsp

[3] https://www.nltk.org/



User:
Can you send me the form for the thesis?

Keywords	Intent
Thesis	Thesis_form
Registration	
Form	
Link	
Credits	Thesis_ECTS
ECTS	
Thesis	

Intent	Response
Thesis_form	Here is the link of the Bachelor registration form file for [BA/MA] students: [link]
Thesis_ECTS	The [BA/MA] thesis is worth 18 ECTS.

Chatbot:
Here is the link of the Bachelor registration form file for [BA/MA] students: [link]

Figure 3.4: Idea of internal functioning of the IfI chatbot based on [46]

3.5.3 Practical Solutions

Although this thesis does not have the concrete implementation of the chatbot as a goal, it is important to at least show a basis on which to build in the future. In the sections above, various criteria have been listed to which the chatbot would have to pay attention. Some of these can already be employed for making an initial filtering of the platforms that can be used to create the chatbot. However, it should be pointed out that many decisions that have been made above are not enough to decide on certain aspects, which can only be verified in practice. For example, the fact that the chatbot has to extract variables from websites is a great unknown, since it is not known how the dataset can be uploaded on the different platforms until it is actually tried. For this reason, for the moment only the available concrete element, namely the question of integration, should be taken into consideration. In this way, we can select the applications that can be integrated in a website or in Microsoft Teams.

In the table 3.2, some platforms that meet the integration requirements are indicated. All these options work both for Windows, macOS and Linux.

Botpress is an open-source developers friendly platform for building, managing and deploying chatbots. NLP methods such as intent recognition, spell checking and entity extraction are built-in. It also offers an SDK and a code editor to expand the chatbot's capabilities. It can be integrated in many channels such as MS Teams, Slack, Telegram, Facebook or in an embedded web chat [7].

Dialogflow is a Google-owned NLU platform that is suitable for designing both text- and voice-based chatbots. It provides two main services: Dialogflow ES, for designing simple agents, and Dialogflow CX, for very large or complex ones. It can be integrated in mobile apps, websites, devices and many others [15].

Microsoft Bot Framework offers the possibility to build bots in many different ways. The Bot Framework and the Azure Bot Service provide all the necessary tools for building an intelligent chatbot that can handle Q&A, understand natural language and use speech. Particularly useful is the SDK, which is extensible and available in Python, Java, C# and JavaScript. The Bot Framework

Platform	Language	Implementation method	Integration
Botpress	JavaScript	Framework Graphical Tools	Website, Teams
Dialogflow	No-code	Own Framework	Website
Microsoft Bot Framework	Composer: TypeScript, SDK: others	Bot Framework Composer (No-code), SDK for Python, Java, C# and JavaScript, Power Virtual Agents	Website, Teams
Pandorabots	AIML	Own Framework, SDKs for Python, Java and NodeJS	Website
Python + NLK	Python	Python	Website
Rasa	Python	Python	Teams

Table 3.2: Possible practical solutions for the IfI chatbot implementation

Composer is built on the SDK and is an IDE. Thanks to the visual interface it allows to easily model the conversational experience. It can also be extended with code if more complex activities should be performed. Finally, there is the possibility of creating a chatbot with "Power Virtual Agents". This is suitable for creating chatbots that answer general questions that are usually asked by customers or employees. It is available as a standalone web application or as a direct application in MS Teams. The integration possibilities of Microsoft Bot Framework are numerous: Facebook, Telegram, Webchat, Alexa and Webex are only some examples [34].

Pandorabots is an online web service which, exactly as the A.L.I.C.E. chatbot (see 2.1), uses the Artificial Intelligence Markup Language (AIML). It can be integrated in websites [37].

As already mentioned in 3.5.2 there is still the possibility to create a chatbot with Python and NLTK packages. This would work for the integration in websites if web frameworks that are written in Python, such as Django or Flask, are used. Of course, this method requires more effort in terms of implementation but it is highly customizable.

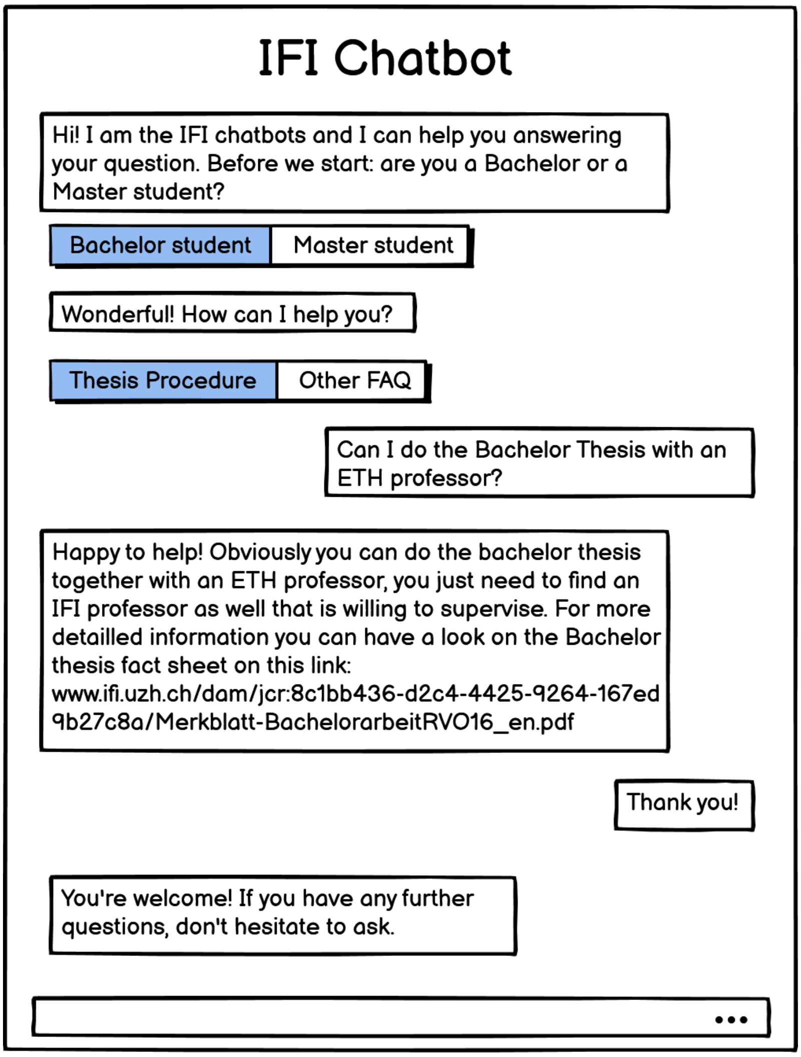
Rasa is thought of as an agent to improve customer experience and has an Open Source version (Rasa Open Source) and a premium one (Rasa X). Rasa Open Source is a machine learning framework suitable both for text-based and voice-based interactions. The focus is put on message understanding and messaging channels/APIs connections. Rasa X is built on top of Rasa Open Source and has the focus on Conversation-Driven Development (CDD), namely the process of remembering users' insights for improving the AI [42].

3.5.4 Mockups

After defining the requirements of the chatbot, deriving its main design decisions and reflecting on aspects such as its functioning and practical implementation solutions, a possible user interface can be sketched.

In figure 3.5 a successful conversation with the chatbot is presented. It should be noticed how the chatbot includes both input given by buttons and text (FR1, FR9). This helps to reduce the complexity of the conversation where it is not needed (NFR1-2).

It is obvious that it is too optimistic to think that the conversation will always produce the desired results. Possible failure scenario are that the students would look for information in the



Hi! I am the IFI chatbots and I can help you answering your question. Before we start: are you a Bachelor or a Master student?

Wonderful! How can I help you?

Can I do the Bachelor Thesis with an ETH professor?

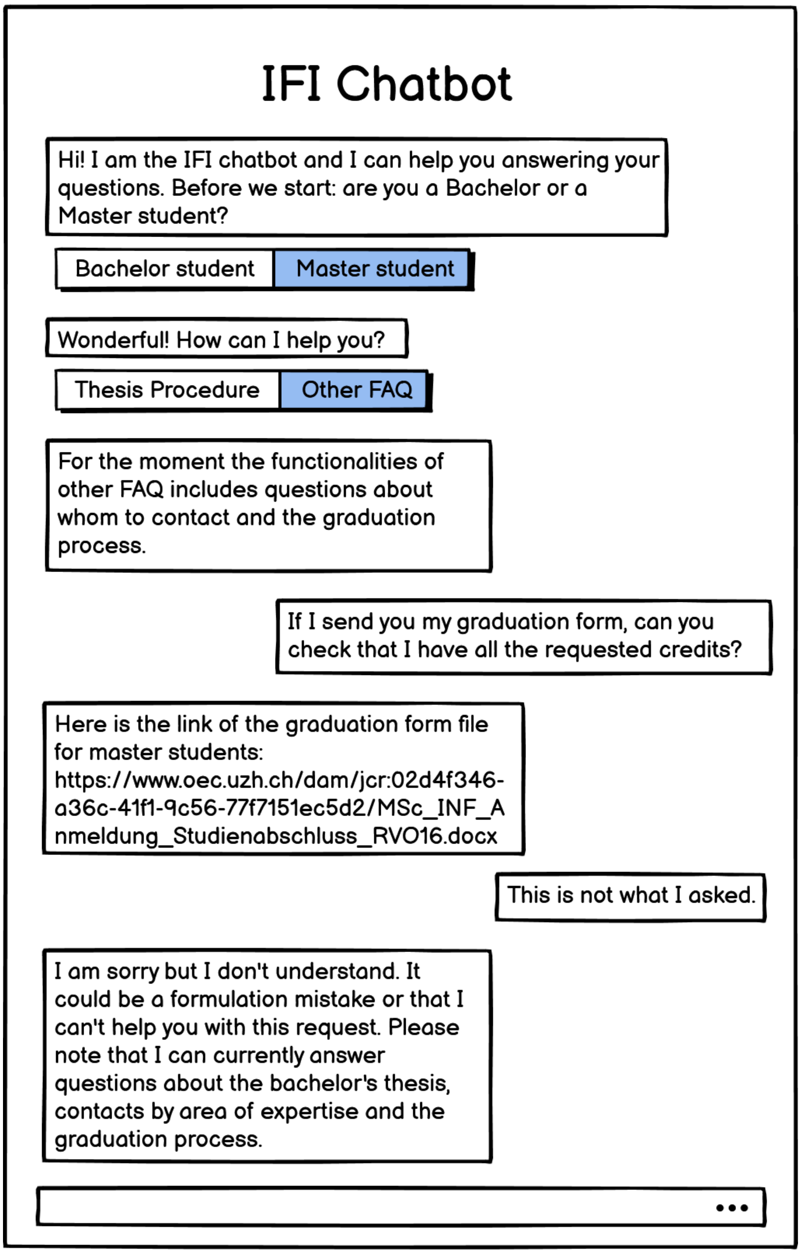
Happy to help! Obviously you can do the bachelor thesis together with an ETH professor, you just need to find an IFI professor as well that is willing to supervise. For more detailled information you can have a look on the Bachelor thesis fact sheet on this link: www.ifi.uzh.ch/dam/jcr:8c1bb436-d2c4-4425-9264-167ed9b27c8a/Merkblatt-BachelorarbeitRVO16_en.pdf

Thank you!

You're welcome! If you have any further questions, don't hesitate to ask.

Figure 3.5: Example of a possible successful conversation with the IfI chatbot

wrong sections, that the information does not exist or that it is incorrect/incomplete. An example of a failure scenario is presented in figure 3.6 in which the chatbot can not satisfy the user's request and it respond with possible explanations of why this could be the case.



Hi! I am the IFI chatbot and I can help you answering your questions. Before we start: are you a Bachelor or a Master student?

Wonderful! How can I help you?

For the moment the functionalities of other FAQ includes questions about whom to contact and the graduation process.

If I send you my graduation form, can you check that I have all the requested credits?

Here is the link of the graduation form file for master students: https://www.oec.uzh.ch/dam/jcr:02d4f346-a36c-41f1-9c56-77f7151ec5d2/MSc_INF_Anmeldung_Studienabschluss_RVO16.docx

This is not what I asked.

I am sorry but I don't understand. It could be a formulation mistake or that I can't help you with this request. Please note that I can currently answer questions about the bachelor's thesis, contacts by area of expertise and the graduation process.

Figure 3.6: Example of a possible unsuccessful conversation with the IfI chatbot

Chapter 4

Conclusion

In this thesis, chatbots were discussed. The theory about chatbots in 2 was useful as a basis for understanding a more specific case based on the Department of Informatics at the University of Zurich (IfI) student office.

There was a need to dig deep to compare different hypotheses and think about every little detail. Nevertheless, the interview helped in identifying the set of activities for which the performance of the student office has a higher chance of being boosted by a well designed chatbot. Reflecting on how a chatbot can be useful in different tasks [47], it is clear how, today, these systems have an enormous potential in this space. To quote a sentence from the interview "Chatbots are patient", and can be an optimal way to improve productivity by avoiding some monotonous and repetitive activities. Furthermore, they also provide immediate responses to the users, who would not need to wait several days anymore.

The specific areas of interest identified for the chatbot consisted in the answering of questions about the thesis, the graduation procedure and the indication of the contact information and responsibilities of each department. With these as a starting point, a requirements definition was created. This also included the specification of user stories, which in turn allowed for the definition of user expectations. Moreover, both functional requirements, describing what the system should do, and non-functional requirements, specifying how the system should operate, were defined. This allowed for the derivation of more specific decisions concerning the design of the chatbot. The application of a Software Engineering process was essential in order to derive a design capable of meeting all the constraints of the student office. The fundamental design decisions that have been made will be repeated below.

As described in 3.5.1, the chatbot for the IfI is of closed domain type and primarily information-oriented but with a task-oriented facet. The service provided is interpersonal, whilst the interaction mode is both text-based and button-based and the response generation is rule-based. However, if the scope of the chatbot is going to be expanded, a retrieval-based method could also be considered, because, with an increasing number of scenarios to define, a rule-based approach would become too expensive [38]. The chatbot accepts inputs and produces outputs in the English language and operates 24/7. Based on the interview findings, ideal integration channels are websites (either on the IfI website or in a separate one) and Microsoft Teams. When a student wants to retrieve the deadline of his Bachelor Thesis, the chatbot is able to interface with the UZH authentication system for extracting the information. The chatbot dataset is half constructed manually and half automatically completed with variables extracted from the IfI and oec website through Web Scraping. This chatbot would be suitable for the staff of the student office because it has a minimum impact on them. Instead of generating new work, it allows them to eliminate the tasks that are repetitive and out of their competence from the workload. As a result, they can better concentrate on their core activities. However, the greatest advantage would be for students, who can obtain valuable information at any time with just a few clicks.

In conclusion, this thesis focused on the accurate design of a rule-based chatbot architecture. For this reason, it is only possible to speculate about the precise performance of the system. An actual implementation would allow us to experiment with different designs and to select specific measures based on the prerequisite that we have a manually built database, where the responses are completed with data extracted through the Web Scraping process. The details of how this type of design can be integrated into a specific framework can only be defined once the technologies for implementation have been chosen. Some possible implementation methods for this particular case are given in the section 3.5.3. These include Botpress, Dialogflow, Microsoft Bot Framework, Pandorabots, Python + NLTK and Rasa. A possible research topic for a future work would be an in-depth investigation on these proposed frameworks for this specific case.

Appendix A

Requirements

A.1 User Requirements

(UR1) Students should have the possibility to get the right information depending on whether they are a Bachelor or Master student.

A.1.1 FAQ

Contacts

(UR2) Given their matter, students should receive the contact of the right person/department to get in touch with.

Degree

(UR3) Students should obtain information about what to do in order to obtain the degree.

(UR4) Students should have the possibility to download the appropriate graduation form.

(UR5) Students should be able to see their degree deadlines.

A.1.2 Thesis Procedure

(UR6) Students should receive information about what the thesis is.

(UR7) Students should receive information about how many credits (ECTS) the thesis is worth.

(UR8) Students should receive information about the thesis duration.

(UR9) Students should receive information about their requirements for starting the thesis.

(UR10) Students should receive information about the possibility of doing the thesis with externals such as ETH professors or companies.

(UR11) Students should have the possibility to download the registration form for the thesis.

(UR12) Students should receive information about where they need to sign the registration form.

(UR13) Students should receive information about how to submit the thesis.

(UR14) Students should have the possibility to check their thesis deadline via authentication.

A.2 Functional Requirements

(FR1) The chatbot should accept text and button inputs.

(FR2) The chatbot should be able to respond both with text and files.

(FR3) The chatbot should accept either English, German or one of the two.

(FR4) The chatbot should work on different OS.

(FR5) The chatbot should function 24/7 without unnecessary interruptions.

(FR6) The chatbot's data should be retrieved from different kinds of websites.

(FR7) The chatbot should also have the possibility to manually insert and modify the data.

(FR8) The chatbot should always indicate the source of where the data is taken from when the data is retrieved from a webpage.

(FR9) The chatbot should suggest possible input to the user when he is confronted with a decision.

(FR10) The chatbot should not go live on IfI's social media platforms (Facebook and Instagram).

A.2.1 FAQ

Contacts

(FR11) The contact information should be extracted from the Advisory Services page on the IfI website.

(FR12) The area of concern for each contact should be extracted from the Advisory Services page on the IfI website.

(FR13) The contact of the dean's office should be extracted from www.oec.uzh.ch.

(FR14) The responsibilities of the dean's office should be extracted from www.oec.uzh.ch.

Degree

(FR15) The link of the graduation form file should be retrieved from www.oec.uzh.ch/en/studies/general/regulations.html.

(FR16) The degree deadlines should be retrieved from www.oec.uzh.ch/en/studies/general/graduation/degreedeadlines.html.

A.2.2 Thesis Procedure

(FR17) The information about the thesis procedure should be extracted from the fact sheet PDF, that is linked in the "While Studying" BA/MA pages on the IfI website.

(FR18) The link of the thesis registration form file should be extracted from the IfI website.

(FR19) The chatbot should extract the deadline for the ongoing theses for each individual student and return it only if an authentication has taken place.

A.3 Non-functional Requirements

(NFR1) The system should be intuitive.

(NFR2) The system should be easy to use.

(NFR3) The system should have short response times.

(NFR4) The system should consider students' privacy.

Appendix B

Interview to Requirements and Design Decisions

#	Q&A	Require-ments	Design Decision	Explanation
1	**Q**: Which are your main stakeholders?	-	Students will be the main user for this use case.	-
	A: We have students (from BA to Doctorate level), postdocs, professors			
2	**Q**: Do you also receive many emails about the internship?	-	Not using the internship as a feature for the chatbot.	-
	A: This has not been a big problem in the past. Regarding the internship we don't receive many questions. Well over corona a was a little bit more, but usually not. But I am thinking that you need something that takes place in the next few months. Because we don’t get requests for the internship at the moment. Because for the next summer we plan only from March onwards. So, maybe that would not be an ideal. You need something where I get quite a few requests on the same topics in a short time.			
3	**Q**: From which stakeholder do most of the questions come from?	UR1	-	Majority of contacts are students so it will be designed for them.
	A: More than half of my contacts are with students. Mostly students that are already studying [...]. But there are a certain number of prospective students that contact us.			
4	**Q**: So at the end is someone else that deals with the actual thing. Like it’s the dean’s office that takes care of these matters?	UR2	Use contacts as a chatbot feature.	Students make confusion on whom to contact.
	A: Sometimes it’s the dean’s office. I forward them to a certain person. But often it’s just links. I am not telling them to contact a person, I tell them where it is written on the website. Then, they can read through. And if they have more questions there is a contact box on the side. But I usually don’t tell them “send an email to this person”.			
5	**Q**: Do students also get confused about who is responsible for what, or who should they ask a particular question?	UR2	Use contacts as a chatbot feature.	-
	A: On our website we have the Advisory Service page saying who is in charge of what. So it’s up to the people who inquire to find out if it is something related to IFI or to study general things. So in the end the people who have the question need to find out whom to contact. And maybe the chatbot would	FR7	-	It is mentioned that the students are confused about the information on the website. So, retrieving information from there

	just ask: “ Is it about general studies or informatics?”. Then the people would be confused: “But I am studying so why do you ask me which one it is?”.			would not be enough. Probably some additional information should be manually inserted in order for the students to understand better.
6	But maybe in connection with the thesis is the graduation process. Often the follow up questions are “Now I want to graduate, can you check my courses?”. And I reply: “No we can't, we are not doing this, if you have specific questions, then you can contact the dean’s office”.	UR3-5	Use graduation procedure as a chatbot feature.	-
7	**Q**: [...] Can you explain what are the Frequently Asked Questions that you receive?	UR3-5	Use graduation procedure as a chatbot feature.	The graduation procedure is part of the Frequently Asked Questions.
	A: Frequently asked questions are typically regarding the thesis. How do I have to hand in my thesis? Where do I find that? Very often I get asked what the admission requirements are for prospective students. On a daily basis I forward them to the dean’s office or to the admission website. This is something that comes in daily. Or questions regarding fact sheets or registration forms that I just forward a link to. Or, how can I get my ECTS credit? Then I forward them the link to the dean’s office website where everything is marked. Can I use a language course for my study? How can I do that? And then there is another link. So, often I just reply by sending a link to the website or a link to the Course Catalogue. Can I use that course for that? You find it in the Course Catalogue. Where is the room? You find it in the Course Catalogue.	UR6-13	Use thesis procedure as a chatbot feature.	The thesis procedure is a topic that causes many questions from the students.
8	**Q**: Let’s take as example the registration for the thesis, and only the communication that you are having with the students. Can you try to describe the kind of questions that you receive from the students?	UR6-14	Use thesis procedure as a chatbot feature.	The thesis procedure is a topic that causes many questions from the students.
	A: What often gets asked is “Can I write the thesis with a professor from ETH?”, “With a company?”. This is a question that is frequently asked and I have to write the same response again and again: “You can if you find an IFI professor as well that is willing to supervise”. Then “Do I need to sign the second page?”. I still write “Yes you have to sign it. Please send it again”.Then, once it is started I confirm that everything is fine and I don’t hear			

	much until before students want to submit. Then they ask “How do I need to submit?”, “All digitally?”, “(How) do I need to submit the code?”, “Don’t you really need a physical copy anymore?”, “What do I have to do afterwards?”, “When is my deadline?”.			
9	**Q**: Could anyone ask you “When is the deadline of my colleagues?”.	UR14, FR19	Authentication is needed for retrieving the thesis deadline.	-
	A: No. I verify if a person sends me an email from their UZH account. Then, I assume it is a student so I answer with the deadline for that name.			
10	**Q**: About the language in which you communicate, is it English or German?	FR3	-	-
	A: Is about half and half.			
11	**Q**: So if you have to generalize this case, are the majority of the questions that you receive answerable with a link?	FR6, FR8, FR11-18	-	Information on how to answer the questions of the cases chosen for the chatbot can be found on the ifi and oec websites.
	A: I am not sure it is the majority, but many.			
12	**Q**: Do you also use other channels such as social media?	FR10	-	-
	A: IFI has a Facebook and Instagram page, but students are not asking questions about studies there. It should be obvious that we are not using our Facebook to inform about our study programs in the way such a study coordinator would.			
13	**Q**: Sorry to generalise you as a role, but you are acting as a search engine. Students are asking you something, you are looking for them for the right link and you are pointing that, right?	-	Suggestion of using either a rule- or retrieval-based approach for the chatbot.	-
	A: In many cases.			
14	With Corona, we switched from delivering a physical copy of the thesis to online delivery. Now we have to start updating the fact sheets and things like that.	-	Use web scraping for extracting data from the website.	The information on the website changes. A dynamic way for extracting the content is needed so that, if information changes the chatbots data is updated accordingly.
15	**Q**: And you also mentioned teams before. Do you get many questions there?	-	Considering Microsoft Teams as a	It could be a good way to separate the role of the administrative
	A: Not so much anymore. When we were working			

	in the home-office and all the students were home, yeah. They used it when they wanted a quicker answer than they would expect from email. But it's more informal. I don't like that because then you have it in that chat and you cannot search for it in your mailbox. My mailbox is also kind of an archive. Sometimes they tell me: "A year ago I asked that and now I have a follow up question".		possible candidate for the chatbot integration.	assistant from the chatbot.

Bibliography

[1] Ahmad Abdellatif, Diego Costa, Khaled Badran, Rabe Abdalkareem, and Emad Shihab. Challenges in chatbot development: A study of stack overflow posts. In *Proceedings of the 17th International Conference on Mining Software Repositories*, page 174–185, New York, 2020. Association for Computing Machinery. doi: .

[2] Eleni Adamopoulou and Lefteris Moussiades. Chatbots: History, technology, and applications. *Machine Learning with Applications*, 2:1–18, 2020. doi: .

[3] Eleni Adamopoulou and Lefteris Moussiades. An overview of chatbot technology. In *Artificial Intelligence Applications and Innovations*, pages 373–383, Cham, 2020. Springer International Publishing. doi: .

[4] Daniel Adiwardana, Minh-Thang Luong, David R. So, Jamie Hall, Noah Fiedel, Romal Thoppilan, Zi Yang, Apoorv Kulshreshtha, Gaurav Nemade, Yifeng Lu, and Quoc V. Le. Towards a human-like open-domain chatbot. *ArXiv*, 2020.

[5] Aravind Mohanoor Aravindmc. How to bulk upload intents from a csv file in dialogflow, 2020. , Last accessed on 2021-10-20.

[6] Jacob Aron. How innovative is apple's new voice assistant, siri? *New Scientist*, 212(2836):24, 2011. doi: .

[7] Botpress. About botpress ·: Developer's guide, 2021. , Last accessed on 2021-11-04.

[8] Petter Bae Brandtzaeg and Asbjørn Følstad. Why people use chatbots. In *Internet Science*, pages 377–392, Cham, 2017. Springer International Publishing. doi: .

[9] Massimo Canonico and Luigi De Russis. A comparison and critique of natural language understanding tools. *Cloud Computing*, 2018:110–115, 2018.

[10] Daniel Carlander-Reuterfelt, Álvaro Carrera, Carlos A. Iglesias, Óscar Araque, Juan Fernando Sánchez Rada, and Sergio Muñoz. Jaicob: A data science chatbot. *IEEE Access*, 8:180672–180680, 2020. doi: .

[11] Rollo Carpenter. About the jabberwacky ai, 2005. , Last accessed on 2021-09-21.

[12] Hongshen Chen, Xiaorui Liu, Dawei Yin, and Jiliang Tang. A survey on dialogue systems. *ACM SIGKDD Explorations Newsletter*, 19(2):25–35, 2017. doi: .

[13] K. M. Colby. *Human-Computer Conversation in A Cognitive Therapy Program*, pages 9–19. Springer US, Boston, 1999. doi: .

[14] Antonella De Angeli and Rollo Carpenter. Stupid computer! abuse and social identities. In *Proceedings of Abuse: The darker side of Human-Computer Interaction*, pages 19–25, 2005.

[15] Dialogflow. Dialogflow documentation | google cloud, 2021. , Last accessed on 2021-11-04.

[16] Octavia Efraim, Vladislav Maraev, and João Rodrigues. Boosting a rule-based chatbot using statistics and user satisfaction ratings. In *Artificial Intelligence and Natural Language*, pages 27–41, Cham, 2018. Springer International Publishing.

[17] FileFormat. Csv file format, 2019. , Last accessed on 2021-10-25.

[18] Jake Frankenfield. Chatbot, 2020. , Last accessed on 2021-04-19.

[19] Ho Thao Hien, Pham-Nguyen Cuong, Le Nguyen Hoai Nam, Ho Le Thi Kim Nhung, and Le Dinh Thang. Intelligent assistants in higher-education environments: The fit-ebot, a chatbot for administrative and learning support. In *SoICT 2018: Proceedings of the Ninth International Symposium on Information and Communication Technology*, pages 69–76, New York, 2018. Association for Computing Machinery. doi: .

[20] Matthew B. Hoy. Alexa, siri, cortana, and more: An introduction to voice assistants. *Medical Reference Services Quarterly*, 37(1):81–88, 2018. doi: .

[21] Minlie Huang, Xiaoyan Zhu, and Jianfeng Gao. Challenges in building intelligent open-domain dialog systems. *ACM Transactions on Information Systems*, 38(3):1–32, 2020. doi: .

[22] Shafquat Hussain, Omid Ameri Sianaki, and Nedal Ababneh. A survey on conversational agents/chatbots classification and design techniques. In *Web, Artificial Intelligence and Network Applications*, pages 946–956, Cham, 2019. Springer International Publishing. doi: .

[23] Srini Janarthanam. *Introduction*, page 7–21. Packt Publishing, Birmingham, 2017.

[24] Rashid Khan and Anik Das. *Business and Monetization*, pages 91–106. Apress, Berkeley, 2018. doi: .

[25] Rashid Khan and Anik Das. *Introduction to Chatbots*, pages 1–11. Apress, Berkeley, 2018. doi: .

[26] Tod Kiryazov. Introducing insurify bot for facebook messenger, 2020. , Last accessed on 2021-10-25.

[27] Pavel Kucherbaev, Alessandro Bozzon, and Geert-Jan Houben. Human-aided bots. *IEEE Internet Computing*, 22(6):36–43, 2018. doi: .

[28] Richard Lawson. *Scraping the data*. Packt Publishing Ltd, Birmingham, 2015.

[29] Susan Li. Named entity recognition with nltk and spacy, 2018. , Last accessed on 2021-11-01.

[30] Lue Lin, Luis Fernando D'Haro, and Rafael Banchs. A web-based platform for collection of human-chatbot interactions. In *Proceedings of the Fourth International Conference on Human Agent Interaction*, page 363–366, New York, 2016. Association for Computing Machinery. doi: .

[31] Abbas S. Lokman and Jasni Mohamad Z. Extension and prerequisite: An algorithm to enable relations between responses in chatbot technology. *Journal of Computer Science*, 6:1212–1218, 2010. doi: .

[32] Saurabh Mathur and Daphne Lopez. A scaled-down neural conversational model for chatbots. *Concurrency and Computation: Practice and Experience*, 31:1–10, 2019. doi: .

[33] Michael McTear, Zoraida Callejas, and David Griol. *The Conversational Interface: Talking to Smart Devices*. Springer Publishing Company, Incorporated, 1st edition, 2016.

[34] Microsoft. Azure bot service documentation - bot service, 2021. , Last accessed on 2021-11-04.

[35] Ketakee Nimavat and Tushar Champaneria. Chatbots: An overview. types, architecture, tools and future possibilities. *International Journal for Scientific Research & Development*, 5(7):1019–1024, 2017.

[36] Eshwitha Sai Palakur. Applications of string matching algorithms, 2020. , Last accessed on 2021-11-01.

[37] Pandorabots. About pandorabots, 2021. , Last accessed on 2021-11-04.

[38] Zhenhui Peng and Xiaojuan Ma. A survey on construction and enhancement methods in service chatbots design. *CCF Transactions on Pervasive Computing and Interaction*, 1:204–223, 2019. doi: .

[39] PostFinance. Postfinance becomes first swiss financial institution to launch chatbot, 2017. , Last accessed on 2021-08-21.

[40] Kiran Ramesh, Surya Ravishankaran, Abhishek Joshi, and K. Chandrasekaran. A survey of design techniques for conversational agents. In *Information, Communication and Computing Technology*, pages 336–350, Singapore, 2017. Springer Singapore. doi: .

[41] Bhavika R. Ranoliya, Nidhi Raghuwanshi, and Sanjay Singh. Chatbot for university related faqs. In *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pages 1525–1530, Udupi, 2017. doi: .

[42] Rasa. Developer documentation portal, 2021. , Last accessed on 2021-11-04.

[43] Nudtaporn Rosruen and Taweesak Samanchuen. Chatbot utilization for medical consultant system. In *2018 3rd Technology Innovation Management and Engineering Science International Conference (TIMES-iCON)*, pages 1–5, Bangkok, 2018. doi: .

[44] Xavier Sánchez-Díaz, Gilberto Ayala-Bastidas, Pedro Fonseca-Ortiz, and Leonardo Garrido. A knowledge-based methodology for building a conversational chatbot as an intelligent tutor. In *Advances in Computational Intelligence*, pages 165–175, Cham, 2018. Springer International Publishing. doi: .

[45] Anand V. Saurkar, Kedar G. Pathare, and Shweta A. Gode. An overview on web scraping techniques and tools. *International Journal on Future Revolution in Computer Science & Communication Engineering*, 4(4):363–367, 2018.

[46] Usman Shahid. Blog: Building a rule-based chatbot in python, 2020. , Last accessed on 2021-10-20.

[47] Bayan Shawar and Eric Atwell. Chatbots: Are they really useful? *LDV-Forum*, 22(1):29–49, 2007.

[48] Amir Shevat. *Designing Bots: Creating Conversational Experiences*. O'Reilly Media, Sebastopol, 2017.

[49] Abhishek Singh, Karthik Ramasubramanian, and Shrey Shivam. *Processes in the Banking and Insurance Industries*, pages 1–18. Apress, Berkeley, 2019. doi: .

[50] Jagdish Singh, Minnu Joesph, and Khurshid Jabbar. Rule-based chabot for student enquiries. *Journal of Physics: Conference Series*, 1228:1–5, 2019. doi: .

[51] Albert Smith. Understanding architecture models of chatbot and response generation mechanisms - dzone ai, 2020. , Last accessed on 2021-10-20.

[52] Rob Sobers. What is oauth? definition and how it works, 2018. , Last accessed on 2021-11-01.

[53] Amrita S. Tulshan and Sudhir Namdeorao Dhage. Survey on virtual assistant: Google assistant, siri, cortana, alexa. In *Advances in Signal Processing and Intelligent Recognition Systems*, pages 190–201, Singapore, 2019. Springer Singapore. doi: .

[54] A. M. TURING. I.—COMPUTING MACHINERY AND INTELLIGENCE. *Mind*, LIX(236):433–460, 1950. doi: .

[55] W3Schools. Javascript tutorial, 2021. , Last accessed on 2021-10-25.

[56] W3Schools. Xml tutorial, 2021. , Last accessed on 2021-10-25.

[57] Richard S. Wallace. *The Anatomy of A.L.I.C.E.*, pages 181–210. Springer Netherlands, Dordrecht, 2009. doi: .

[58] Hao Wang, Zhengdong Lu, Hang Li, and Enhong Chen. A dataset for research on short-text conversations. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 935–945, Seattle, 2013. Association for Computational Linguistics.

[59] Joseph Weizenbaum. Eliza—a computer program for the study of natural language communication between man and machine. *Communications of the ACM*, 9(1):36–45, 1966. doi: .

[60] Tsung-Hsien Wen, David Vandyke, Nikola Mrksic, Milica Gasic, Lina M. Rojas-Barahona, Pei-Hao Su, Stefan Ultes, and Steve Young. A network-based end-to-end trainable task-oriented dialogue system, 2017.

[61] Eric Hsiao-Kuang Wu, Chun-Han Lin, Yu-Yen Ou, Chen-Zhong Liu, Wei-Kai Wang, and Chi-Yun Chao. Advantages and constraints of a hybrid model k-12 e-learning assistant chatbot. *IEEE Access*, 8:77788–77801, 2020. doi: .

[62] Yu Wu, Wei Wu, Chen Xing, Ming Zhou, and Zhoujun Li. Sequential match network: A new architecture for multi-turn response selection in retrieval-based chatbots. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, volume 1, Vancouver, 2017. Association for Computational Linguistics. doi: .

[63] XOR. Xor | recruiting automation software for faster hiring, 2021. , Last accessed on 2021-04-19.

[64] Zhou Yu, Ziyu Xu, Alan W Black, and Alexander Rudnicky. Chatbot evaluation and database expansion via crowdsourcing. In *Proceedings of the chatbot workshop of LREC*, volume 63, page 102, 2016.

[65] Jennifer Zamora. I'm sorry, dave, i'm afraid i can't do that: Chatbot perception and expectations. In *Proceedings of the 5th International Conference on Human Agent Interaction*, page 253–260, New York, 2017. Association for Computing Machinery. doi: .

[66] Darius Zumstein and Sophie Hundertmark. Chatbots - an interactive technology for personalized communication, transactions and services. *IADIS International Journal on WWW/Internet*, 15(1):96–109, 2017.