Universidade de Aveiro
2022

Tiago Jorge
Costa Correia

Exploração de Assistentes Conversacionais para Disponibilizar Informação de Acessibilidade em Turismo

Exploring Conversational Assistants to Provide Accessibility Information in Tourism

Universidade de Aveiro
2022

Tiago Jorge
Costa Correia

Exploração de Assistentes Conversacionais para Disponibilizar Informação de Acessibilidade em Turismo

Exploring Conversational Assistants to Provide Accessibility Information in Tourism

"*The greatest challenge to any thinker is stating the problem in a way that will allow a solution*"

— Bertrand Russell

Universidade de Aveiro
2022

Tiago Jorge
Costa Correia

Exploração de Assistentes Conversacionais para Disponibilizar Informação de Acessibilidade em Turismo

Exploring Conversational Assistants to Provide Accessibility Information in Tourism

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia de Computadores e Telemática, realizada sob a orientação científica do Doutor Samuel de Sousa Silva, Professor auxiliar do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro, e do Doutor António Joaquim da Silva Teixeira, Professor associado com agregação do Departamento de Eletrónica, Telecomunicações, e Informática
.

This work was developed in the scope of the research project ACTION – Accessible Tourism: Co-Creation of Tourism Experiences Through Web-Based Intelligent System, funded by FEDER, through COMPETE2020 – Programa Operacional Competitividade e Internacionalização (POCI-01-0145-FEDER-030376), and by national funds (OE), through FCT/MCTES (PTDC/EGE-OGE/30376/2017)

o júri / the jury

presidente / president	Professora Doutora Maria Beatriz Alves de Sousa Santos Professora Associada c/ Agregação, Universidade de Aveiro
vogais / examiners committee	Professor Doutor Alberto Manuel Brandão Simões Professor Adjunto, Instituto Politécnico do Cávado e do Ave
	Professor Doutor Samuel de Sousa Silva Professor Auxiliar, Universidade de Aveiro

agradecimentos / acknowledgements

Agradeceço aos meus pais por todo o esforço que fizeram para que fosse possível eu frequentar e acabar o curso.

Agradecer aos meus orientadores, professor Samuel Silva e professor António Teixeira por todas as ajudas e pelo tempo disponibilizado durante a dissertação.

Agradeceço também ao meu primo Nuno Almeida por toda a ajuda que me deu durante todo o curso, sempre disponível para ajudar.

Agradeceço também à Helena, Diana e restantes colegas de curso pela companhia e amizade.

Agradecer à restante família por todos os incentivos e apoio que me deram.

Agradecer, por fim, à Mafalda por toda a compreensão e carinho durante grande parte do curso.

Palavras Chave

assistente conversacional, RASA, turismo acessível, desenvolvimento assente em conversação.

Resumo

Embora hoje o turismo seja uma das atividades económicas mais importantes em todo o mundo, contribuindo com 10,2 % do Produto Interno Bruto (PIB) e 10 % do emprego total gerado, ainda existem vários grupos na nossa sociedade que enfrentam muitas restrições de viagens, por exemplo , pessoas com necessidades especiais. A contribuição do turismo para a promoção de uma sociedade inclusiva implica a adoção de estratégias que promovam a acessibilidade dos destinos turísticos, permitindo a todas as pessoas, independentemente das suas capacidades, usufruir de experiências turísticas. A informação sobre a acessibilidade de diferentes espaços e infraestruturas desempenha um papel significativo na forma como as pessoas com diferentes tipos e níveis de incapacidade podem planear as suas vidas, especialmente as experiências turísticas. A este respeito, existem desafios importantes, não só na recolha de todas as informações relevantes, mas também, e sobretudo, na sua disponibilização às pessoas com necessidades especiais de formas, também elas, acessíveis e adaptáveis às diferentes necessidades e capacidades. A não apresentação dessas informações, geralmente, leva as pessoas a desistirem de experienciar o turismo por não se sentirem confortáveis em fazê-lo num ambiente que possa não estar adaptado às suas necessidades e preocupações. Embora a informação já exista, a quantidade de informação torna complexo o acesso à mesma de uma forma simples e intuitiva. Portanto, ter uma forma mais simples e natural de obter as informações desejadas é um aspecto importante a ser explorado. Nesse contexto, o suporte ao diálogo em linguagem natural, seja por voz ou escrita, pode trazer vantagens como complemento ou alternativa às interfaces mais comuns, como portais ou menus, para apresentar as informações, pois pode trazer, potencialmente, alguma naturalidade e eficiência da comunicação através da fala à interação com estes sistemas. E embora possa ser uma alternativa de interação relevante para todos, pode favorecer aqueles que enfrentam dificuldades para aceder e aprender sistemas mais tradicionais. Neste trabalho apresentamos os resultados para o desenvolvimento e instanciação de um assistente conversacional para Turismo Acessível. Com base na metodologia desenvolvimento assente em conversação e considerando o RASA para o desenvolvimento e o backend RASA X para a instanciação do assistente, a utilização do assistente por especialistas do domínio possibilitou um processo iterativo para o seu refinamento e melhoramento. Atualmente, o assistente permite responder a perguntas sobre hotéis, museus e locais de interesse, juntamente com informações de acessibilidade relacionadas com esses espaços e fornece bases promissoras para futuras evoluções.

Keywords conversational assistant, RASA, accessible tourism, conversation-driven development.

Abstract

Although nowadays tourism is one of the most important economic activities worldwide, contributing 10.2% of the Gross Domestic Product (GDP) and 10% of the total employment generated, there are still several groups in our society that face many travel constraints, e.g., persons with disabilities (PwD). The contribution of tourism to the promotion of an inclusive society implies the adoption of strategies that promote the accessibility of tourism destinations, allowing all people, regardless of their abilities, to enjoy touristic experiences. Information on the accessibility of different spaces and infrastructures plays a significant role in how people with different types and levels of disability can plan their lives, especially their touristic experiences. In this regard, important challenges exist, not only in gathering all the relevant information, but also, and most of all, making it available to PwD in ways that are also accessible and adaptable to different needs and abilities. The failure to present this information usually leads people to give up living tourist experiences because they do not feel comfortable to do it in an environment that may not be adapted to their needs and concerns. Although it already exists, the amount of information makes it complex to reach the information in a simple and intuitive way. Therefore, having a more simple and natural way to get to the information that you want is an important aspect to explore. In this context, supporting dialogue in natural language, whether by voice or writing, can bring advantages as a complement or an alternative to the more common interfaces, such as portals or menus, to present the information, as it can potentially bring some of the naturalness and efficiency of speech communication to the interaction with these systems. And while it can be a relevant interaction alternative for all, it can favor those that might face difficulties in accessing and learning more traditional systems. In this work we present the results for the design, development and instantiation of a conversational assistant for Accessible Tourism. Based on a conversation-driven development methodology and considering RASA for the development and RASA X backend for the deployment of the assistant, the use of the assistant by domain experts enabled an iterative process for its refinement and improvement. At its current stage of development, the assistant enables querying about hotels, museums and locals of interest suggestions along with related accessibility information and provides promising grounds for further evolutions.

Contents

CHAPTER 1

Introduction

1.1 Context and Motivation

It is estimated that "30% of the population will have access requirements at any point in time, and most people will likely have a disability at some stage during their life" [1]. So,it is likely that this will lead to a higher number of individuals who will have needs for assistance and special requirements to access environments, services or products in the near future. And one domain in which these aspects have been studied pertains Tourism.

Because **people with disabilities (PwD)** have needs that other people don't have, information on accessibility of different spaces and infrastructures plays an important role in how people with different types and levels of disability can plan their lives including their **touristic experiences**, which can play a pivotal role to improve their physical and mental health and contribute to a greater perceived quality of life [2]. But the failure of the locations or facilities fulfilling certain requirements can lead them to give up on a tourist trip because they do not feel comfortable enough to do it alone [3].

1.2 Challenges

The target audience, usually, experience some difficulties in accessing the most common hierarchical menu systems, handling all the information and using graphical interfaces for this purpose. Because access to information is so crucial for PwD regarding tourism offers, dialogue in natural language, whether by voice or writing, can bring advantages as a complement or an alternative to more common interfaces as it makes it easier for PwD to deal with all the information that exists on tourism offers and to find the specific information they are looking for.

In the scope of project ACTION, a project that aims to find solutions to provide accessibility information for PwD in tourism related activities, one of the main goals is to provide accessible tourism for PwD. To achieve this goal it is taken into account several factors that were seen as challenges and that were in the way of providing accessible tourism for PwD, that is, the main difficulties they found when trying to find information about places to visit and hotels to stay, for example, before and during the trip.

- Complementary information spread over several sites
- Not easy to find the information they want
- Too much information which makes it complicated to navigate.
- Overuse of portals that makes everything more complicated
- Lack of information about things that are the biggest concerns of PwD (ramps, sizes, information in braille, etc)
- Misleading or false information
- Lack of feedback of other people in the same situation

In light of these challenges, work needs to be done at different levels, not only to improve the quality of the available information, but also, to provide an easy and efficient way to access the relevant information for each person and context. In this regard, providing a **Conversational Assistant** as an alternative or complement to other forms of interaction can bring some advantages. By talking to a conversational assistant, instead of searching for what they want in a menu or other platform, PwD will find it much easier and faster to access the information they want and they won't have to go through all the information just to get what they are searching for. That leads to the need to explore platforms to develop the assistant.

1.3 Objectives

Considering the challenges identified in the previous section, the work carried out in this dissertation aims to make it easier the experience of obtaining information regarding the accessibility of hotels, museums and other locals for PwD and giving these people all the tools necessary to do it. To accomplish this goal, several objectives are considered:

- Get acquainted with the domain of Accessible Tourism and major difficulties and concerns in this area, and with the overall aspects involved in designing and developing conversational assistants.
- In collaboration with domain experts and prospective users define the requirements for proposing a novel approach to accessing accessibility information.
- Explore a framework as the grounds to propose a solution that tackles the needs and motivations identified.
- Based on an iterative approach, develop and evaluate a proof-of-concept of a conversational assistant for Accessible Tourism

1.4 Publications

The work described in this document has already resulted in 3 publications.

- Teixeira, P., Alves, J., Correia, T., Teixeira, L., Eusébio, C., Silva, S., Teixeira, A. (2021, July). A Multidisciplinary User-Centered Approach to Designing an Information Platform for Accessible Tourism: Understanding User Needs and Motivations. In International Conference on Human-Computer Interaction (pp. 136-150). Springer, Cham.
- Tiago Correia, Samuel Silva, António Teixeira (2021, September). On the Development of a Conversational Assistant for Accessible Tourism. Inforum
- Tiago Correia, Samuel Silva, António Teixeira, Joana Alves, Pedro Teixeira (2021, November). Designing and Instantiating a Conversational Assistant for Accessible Tourism. SSAT21

1.5 Document Structure

The remainder of this document consists of 5 chapters.

- **Chapter 2**: Background and Related work, provides an overview of the barriers PwD usually face when performing tourism activities and also enlights how PwD can take advantage of a user-centered design approach, not only when booking a trip but also during the tourism trip. Last, address the conversational assistant approach: the different options of platforms to develop the assistant, including the chosen option, RASA, the main modules of an assistant and the RASA architecture.
- **Chapter 3**: Personas, Scenarios and Requirements, addresses the methodology followed to obtain a user-centered design technological solution that benefit and help PwD in tourism related activities.
- **Chapter 4**: Development in RASA 2.0, first overview and then details the process to develop an assistant in RASA. Includes the definition of the different pieces of the assistant, like intents, actions, dialogue policies: a recipe to build an assistant in RASA.
- **Chapter 5**: Assistant Development and Deployment, provides the implementation of the first version of the assistant for accessible tourism and the outcomes for testing the assistant that resulted in a second and third version of the assistant.
- **Chapter 6**: Conclusions, provides an overview of the assistant and the future work to be done using the assistant as a tool for accessible tourism.

CHAPTER 2

Background and Related Work

In this chapter the main goal it to get familiarize with accessible tourism: first, the different types of disability and how that influences tourism activities. After that, it is important to address the barriers PwD face before and during tourism trips, accessing to information concerning the accessibility of hotels, museums and others, due to the lack of suitable information and proper environments that are not prepared to receive people with different disabilities. To continue, it is crucial to explain how a iterative user-centered design approach can benefit PwD in terms of planning and ultimately experience in a better way accessible tourism, trying to think of a solution, using literature as the grounds, for the challenges identified in the previous chapter. To end this chapter and to prepare the next ones, a closer look to conversational assistants: first define and understand conversational assistants, then a short review on the different platforms to develop an assistant and, to end, a closer review on the chosen one, RASA and their main modules in their architecture.

2.1 Accessible Tourism

A disability is a restriction or lack of ability, that results from a impairment, to perform an activity within the range considered normal for a human being [4]. It is estimated that "30% of the population will have access requirements at any point in time, and most people will likely have a disability at some stage during their life" [1]. With that in mind, it is quite probable that this will lead to a much higher number of individuals who have needs for assistance and special requirements to access environments, services or products in the near future. In Portugal, already 18% of the population have some type of disability [5], a number that should increase in the future, so it is important to think about the tourism experience for PwD.

The accessibility of the spaces (hotels, museums) and the accessible information are of utmost importance in order to facilitate the participation of PwD in tourism activities [6]. At first, the accessibility of the physical environments and services are crucial because physical disabilities affects people´s capacity to move around in a normal way [7]. So, is is only normal that some structural constraints as lack of accessibility on hotels, museums and public transports are the biggest concern of PwD when they are planning a tourism trip [8] [9].

Beside physical disabilities, there are also people that have sensor disabilities, whether is affects the vision (blindness if a person can´t see at all or low vision if the lost vision is not corrected with glasses) or the hearing (deafness if a person can´t hear) [10].

So, to people with sensor disabilities, the way the access information is presented is a significant aspect. Normally, the information is visual or in audio formats [10], so for deaf people it is required visual support, like sign language for example, or visual alarms systems to be able to fully understand and communicate with other people [11].

For blind and low-sighted people it is significant that a hotel or museum have tactile support, like braille, or audio, like screen readers to better understand the information provided. Alongside with how to understand the information provided, that can be solve using braille or screen readers, blind and low-sighted people also experience mobility difficulties. Because of that, physical accessibility is also important so they can move around in safety [12]. People with sensor disabilities share the concern of people with physical disabilities of structural constraints of physical spaces but also have troubles with communication, because sometimes they don't understand the information provided, if there aren't accessible ways to read or access the information and they depend on friends or relatives in order to communicate with others [12].

After taking into account, physical and sensor disabilities, it remains to address intellectual disability. An intellectual disability is characterised by someone who have an IQ below 70, as well as difficulty with normal daily living tasks such as self-care, communication, and socialisation. People with an intellectual disability may process information more slowly, find communication and daily living skills hard, and also have difficulty with abstract concepts such as money and time. This disability can affect, not only, intellectual functions as learning and problem-solving capacities but also social behavior as perform simple daily living activities [1]. So, the major problem for people with intellectual disabilities are processing information and adapting their social behavior. In order to diminish this issue, use of simple language and pictograph writing can help in communication and personal assistance to help in daily activities like schedules and use of money are the most common requirements for people with intellectual disability [13] [14].

But, beyond all these constraints and difficulties, PwD wants, like everyone else, to engage more in social activities, especially tourism activities. The only difference is that, PwD need better conditions and more accessibility information in order to feel comfortable and participate more in activities. Unlike most of us, PwD need to know some accessibility information before they visit a museum or choose an hotel so stay during the trip. That's why it's important, not only to have adequate environments and services but also information about those things in order to facilitate the selection process before and during the tourism activities. With that, we can assure to all people, regardless of their disability, several ways

[1] https://www.aruma.com.au/about-us/about-disability/types-of-disabilities/types-of-intellectual-disabilities/

to access and fully enjoy tourism experiences.

2.2 Barriers and Support Technologies

It is common sense that PwD face many constraints/barriers that doesn't even cross the mind of people without any type of disability. PwD face these obstacles when attempting to participate in a tourism experience [15]. There are three major areas that we can place all these constraints: physical access, attitudinal barriers and lack of information [6].

Regarding physical access, it is a major barrier for PwD the possibility of finding environments that are inaccessible [16], like public transports, accommodation facilities and even attractions that are potentially the center of concerns in a tourism trip. Regarding attitudinal barriers, despite the change in public awareness, negative attitudes are still a barrier to accessible tourism [17]. The negative attitudes of tourism staff towards PwD normally prevent to provide these people with correct and reliable information, information that is vital while planning the trip, choosing which places to visit and which accommodation facility to stay [18]. The abusive attempts to sell package tours is another problem because most package doesn't suits the requirements and needs of disabled people, which leads to a questions: whether travel agencies are facilitating or being another barrier to accessible tourism [19].

A lot of technologies have been developed in order to assist and help improving the experience for PwD during tourism and even non-tourism activities, like visits to a museum or just to help finding the information they want in a platform. Of all the developed technologies, it is important to address the ones that have the most impact in tourism and that facilitates tourism experiences for PwD.

Usually, monitors and keyboard are the communication channels between humans and computers. But for people with disabilities as low vision or cerebral palsy is hard or even impossible to use those [20]. So, for blind or low vision people there are some alternative: to use screen readers with sophisticated speech synthesizers, a tactical mode using refreshable Braille displays, enlarge text in browsers or even use non graphical browsers [21]. Regarding museums and hotels, there are a lot of technologies that can be used in order to improve the experience for people with sight and hearing disabilities. Some of the technologies include audio guides, touch tours, information in Braille and interpretation in sign language [22].

2.3 Access accessibility information: Developing Accessible Iterative Systems

Considering the different barriers (physical access, attitudinal barriers and lack of information) addressed on the last section, one of the most important one that define how PwD plan and ultimately live their touristic experience is related to how they access information regarding accessibility of spaces, like hotels, museums and others, named accessibility information [6].

Because of their special needs, PwD when planning a travel, unlike most of the travelers without disabilities, spend a lot of time searching for more detailed information. These people need specific accessibility information regarding their special needs, which is way different than the information searched by people without disabilities. And, often, when they can´t find this information they don´t feel as attracted to engage in tourism activities and diminish the market potential of tourism [23].

Besides having assisting technologies, the ones addressed in last section, like interpretation in sign language or information displayed in Braille, it is also really important that PwD can easily find out if a certain hotel or museum does have those assisting technologies. For example, if a person with physical disabilities can´t find accessibility information regarding a certain hotel nearby the city he will travel to, he probably won´t choose that hotel to stay. But if that happens to each and every hotel in that same city, the most likely is that he will give up on the trip, because he doesn't feel comfortable enough to stay at a hotel where he doesn't know if he will be able to move around in a safe and cozy way. Likewise, if a person with sensor disabilities (blind) is planning a leisure trip to Lisbon and can´t find information about audio-guides in every museum in the city, she probably won´t travel to that place as she doesn't know if she can visit and enjoy that same visit of a museum.

Like these two, there are so many more examples of constraints and barriers PwD find if they can´t access accessibility information of spaces. The two examples above shown a problem that makes the tourism experience for PwD much harder than for people without disabilities, that doesn't even think about that things.

A person without disabilities, when planning a trip, normally wants to know prices of hotels and opening and closing hours of museums. And the lack of that information, doesn't, normally, stop us from traveling. On the contrary, a PwD needs a lot more information of spaces in order to prepare a trip and the lack of information can stop them from travel, and often does. A lot of the technology solutions doesn't cover this area, and the ones that does have all the information available run into other problem: the large ammount of data that makes the search for the specific information they want too difficult. It is required some type of interaction to make it easier to deal with all the information available.

It is, in this regard, important to propose systems that are accessible for a wide range of audiences and abilities, and also, that can deal with all the information available and, at the same time, not overloading the user with all the information at once.

Of the different modes of interactions, there has been a strong development in voice, because it is the easier and most intuitive way of communicating. Because of that, there have been a recent growth in conversational agents, also supported by two factors: conversational agents as systems that can deliver information to a large amount of users simultaneously and the great advances in the field of artificial intelligence, focused on natural language software [24].

2.4 Conversational Assistants

Before thinking about the development of the conversational assistant, it is mandatory to think about the possible dialogues between the user and the assistant. Dialog acts can be used in order to expose the first basic functionalities of the technological solution chosen, the conversational assistant. In the scope of a conversational dialog, a dialog act is an utterance, like a question, statement or a request for an action, that serves a function in the dialog

[25]. The dialog acts can presented some possibilities of interactions in a conversation style between an user and the system itself. Using dialog acts, it becomes clearer the possible interactions between the user and the assistant, what the users wants to know, how the user can ask for a specific information and what responses the assistant should give to the user. Also, the information the assistant needs to keep track so it is aware of the context of the conversation can be withdrawn from dialog acts. In summarize, dialog acts can be used as a first sketch of the conversations that will be possible to have in the future with the conversational assistant. Below, it is an example of a dialog act for a conversational assistant that provides the horoscope.

Table 2.1: Illustrative example of a dialogue devised for an assistant that provides the horoscope.

Sentence	Intents/Responses	Entities
User: Olá assistente!	greeting	
Assistant: Bom dia. Quer saber o seu horóscopo?	utter_greet	
U: Sim, quero saber o horóscopo	request_horoscope	sign=null
A: Qual é o seu signo?	ask_for_sign	sign=null
U: Sou leão	inform_sign	sign=leo
A: O horóscopo para o leão é: ...	inform_horoscope	
U: Obrigado, adeus.	goodbye	
A: Não queres saber mais nada? Até uma próxima, adeus.	utter_goodbye	

Dialog acts are really helpful when attempting to develop a conversational assistant, because it gives a first impression of the features the assistant is required to have in order to carry out the requests of the users.

Conversational agents exploit natural language technologies to engage users in text-based information-seeking and task-oriented dialogues for a broad range of applications [24]. Conversational agents are catalyzed by natural language input by the users, input by voice, text or both. With the input, the task of the conversational agent is to provide a response based on the input by the user [26]. Once a message is sent by a user, it is processed by the Natural Processing Understanding(NLU) unit that tries to match the message to a user action, also known as intent [27]. It is also the goal of the NLU unit to extract any entities (structured pieces of information in a user message) the messages contains. The NLU's output is used by the Dialog Manager to update the state of the conversation. Once it is updated, dialog policies, the central piece of the dialog manager are triggered in order to decide the next action, meaning, the response of the conversational agent to the user [27]. These are the two main modules of the conversational assistant: NLU module that is responsible to identify the intent and extract entities (if that is the case) in the user's message. After that, the Dialog Manager decides the next response (actions) depending on the state of the conversation.

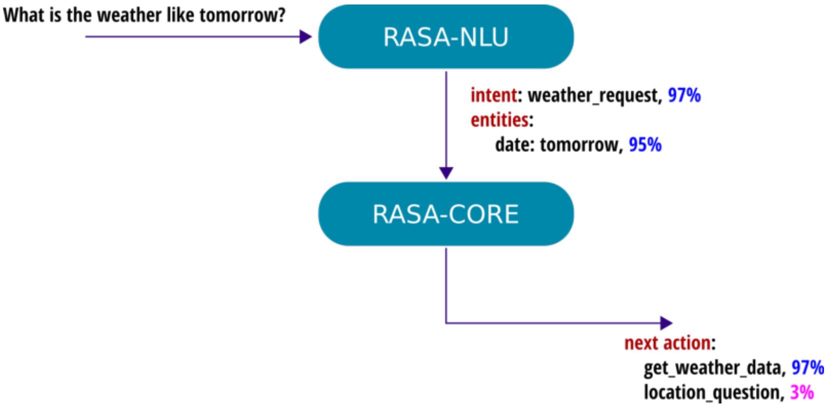


Figure 2.1: Overall explanation of how conversational assistants work. Image obtained from DAITAN.

2.4.1 Recent work on Conversational Assistants in assistive scenarios

As addressed above, a Conversational Assistant can be an important tool and is being used more and more often nowadays because they can evolve to reach a larger number of users than human operators and can provide information in a interactive way. Some examples, that are not related to assistive scenarios, but still use Conversational Assistants as a tool are **DIA** [28] and **Evorus** [29]. DIA is a human-chatbot designed to chat-based interventions for the developing world and was designed for social media, not only to improve sustainability, but also to empower local language interaction to support emergent users. **Evorus** is also a recent developed Conversational Assistant which allows users to talk to Evorus in open domains and the responses are chosen between suggestions offered by crowd workers and the automated systems that have been added to Evorus.

Regarding assistive scenarios, a good example is the recent situation of Covid-19 that rapidly evolved worldwide and alarmed a lot of people that wanted to obtain information regarding this new disease. Several countries developed and used Conversational Agents in order to tackle the needs of providing a lot of information for a lot of people with different requests. As examples are **Theano**, a Conversational Assistant for Covid-19, greek-speaking that informed people with statistics and facts regarding Covid-19, the best health practices and, as well, the latest guidelines for Covid-19 [30]. Another example, developed in Spain is the **Hispabot-Covid19**, the official Spanish conversational assistant, developed by the Spanish government to answer the most frequently asked questions regarding Covid-19. This bot received over 350.000 queries in just three months, which clearly shows that was highly used by people [31]. **Chloe for Covid-19** [32] is also a good example of using Conversational Assistants to provide information in a safe and secure way, keeping the population informed regarding public health situations.

Nevertheless developing CA is a complex process that requires gathering a set of pieces, as described above, and requires the support of dedicated developing frameworks as described in what follows.

2.4.2 Developing Conversational Assistants

The development of a conversational assistant involves the consideration and configuration/-training of a complex set of modules to support, e.g., natural language understanding (NLU) and dialogue management, and several frameworks have been proposed to support developers in one or more of these stages, such as:

- PyDial: an open-source statistical spoken dialogue system that is easy to configurate and easy to extend. Also has domain-independent implementations of the dialogue system, meaning the implementation of all dialogue modules are kept separate from the specification of domain. All functionalities are controlled in a configuration file [33].
- Emora: a social chatbot that looks for interesting information in a independent way and gives subjective opinions about topics she cares. Also, tries to understand things you want and actively engage in conversations for you. It elevates the conversational assistant almost to a level of a friends, which can entertain you and give you company [34].
- Plato: a flexible conversational agent platform written in Python that supports either standard or trained components architecture. Also provides training either online or offline of all conversational AI components. The major focus of Plato is to be easy to understand and debug [35].
- OpenDial: an open-source framework that builds and evaluates spoken dialogue systems. It has been deployed in application domains like human-robot interaction and multimodal driver assistants. It relies on an information-state architecture where the dialogue state acts as a shared memory for all system modules [36].
- Mircrosoft ICECAPS: an open-source natural language repository. It presents a flexible paradigm to construct learning setups. Multitask learning between models that share parameters, a user-friendly data processing pipeline and built-in architectures are some of the capabilities of ICECAPS. It also provides some conversational models that are pre-trained and power an online demo of ICECAPS's framework [37].

Although all five frameworks presented above could be chosen to develop a conversational assistant, in the past few years, the RASA [38] conversational AI platform has gained prominence given its continuous evolution, the seamless integration between the different modules, the low learning curve to obtain a first version of a working assistant and, most importantly, the ability to do so with small amounts of data and supported by an interactive backend for analyzing conversations, annotating data and retraining models.

2.4.3 RASA Framework

RASA is a machine learning framework for developing artificial intelligent chatbots. It is incredibly powerful and is used, all over the world, by developers in order to create contextual assistants. RASA is based on natural language understanding (NLU), so it acts like humans interacting with each other and having in mind not only the context but also what actions are to be taken regarding the current context. Also, like humans, RASA can handle unexpected conversations and recover when user drifts from the conversation path [39].

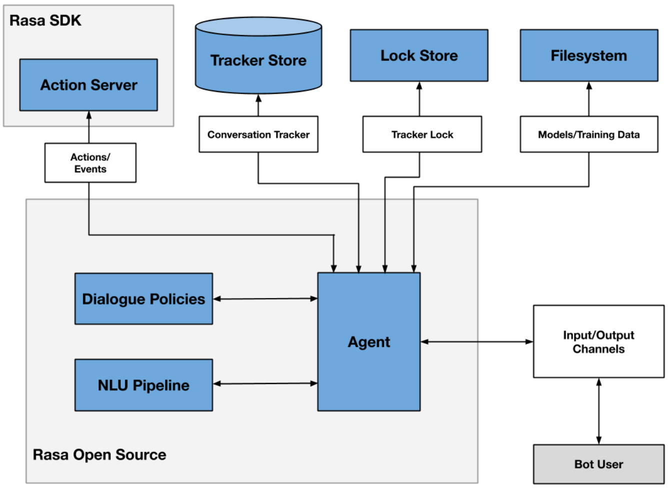


Figure 2.2: Overall architecture of the RASA platform depicting its main modules. Image obtained from RASA.

RASA AI has two primary components: RASA NLU, that is responsible for managing all training data and dialogue management that decides the next action bearing in mind the context, In the figure 2.2, the dialogue management component is described as Dialogue Policies [2]. RASA NLU consists of example user utterances categorized by intent and can include entities. To help identify the correct intent, training data can include regular expressions and lookup tables. To help training the assistant's dialogue management there are stories and rules, that represent possible conversations between the user and the assistant and that are used to train the dialogue management. Unlike rules that are short pieces of conversations that follow, always, the same path and that are used to train RulePolicy, stories can generalize to new and unseen conversation paths. Stories consist in a number of steps (user messages and responses by the assistant) that make up a story. Rules, just like stories, contains a list of steps but additionally have conditions under which that rule can be applied.

Regarding Dialogue Policies, are defined in a config file and the assistants uses them to decide the next action to take in each step of the conversation. There are two types of policies: rule-based, that handles the conversation as a fix behaviour and check the rules file in the training data and machine-learning policies. You can include multiple policies in a single configuration but you need to define the priority levels for each policy in the case more than one policy returns a next action with confidence equals to 1.

The Transformer Embedding Dialogue (TED) Policy is a machine-learning policy and is used to predict the next action to take and to recognize entities. It consists of several transformer encoders that are shared for both tasks.

UnexpecTED Intent Policy and **Memoization Policy** are also machine-learning policies. The first allows your assistant to react to unlikely user turns but should always be used along side other policy because the only action it triggers is the `action_unlikely_intent`.

[2] Because it is a recent framework, in continuous development, the most accurate way of finding information regarding its architecture, features and framework is the online site, available in www.rasa.com

It shares the same model architecture as **TEDPolicy**, the only difference being that instead of learning the best action to be triggered next, if the intent predicted by NLU is supposed to occur given the context, UnexpecTED Intent Policy doesn't trigger any action. Otherwise it triggers the action_unlikely_intent with confidence equals to 1. The second, matches the current conversation with the stories in the training data. If it matches it returns the next action with confidence equals to 1, if not it predicts None with confidence 0. We can change the number of turns we want the policy to take into account, for example one turn if we want the policy to just take into account the previous message sent by the user and the last action performed by the assistant.

The Action Server includes the several type of actions that the assistant should perform after the model predict the correct one and is divided into responses, custom actions, forms and default actions. Responses are used when you want the assistant to send text, buttons, images or something similar to the user. A custom action can run any type of code you want, including database queries, calling an API and more. Forms are a special type of custom action, designed to ask the user for a specific set of information, like booking a restaurant, and it is also called slot filling. Lastly default actions are actions built in the dialogue manager by default. They are automatically predicted based on certain conversation patterns but can be customized in order to personalize the assistant.

Tracker Store is where the conversations of the assistant are stored. You can create your own custom one but, by default, the conversation history is stored in memory. Lock Store allows multiple RASA servers to run in parallel as replicated services. This is possible because RASA uses a ticket lock mechanism to guarantee that incoming messages are processed in the right order and locks conversations when messages are being processed. Regarding Filesystem, you can load your models from three different places after you trained your assistant: you can load it from your local disk, your own HTTP server and from cloud storage. By default, it is loaded from your local disk.

2.5 Iterative User Centered Design

Developing accessible technologies is not only a matter of having several choices for interaction to facilitate the task for users. It is also paramount that the users play a pivotal role in design, developing and evaluating what is proposed so the final product is completely according to the interests of users. The opposite approach can leave users wasting too many time trying to complete a simple task because it is not intuitive and simple for them. User Centered Design refers to "design process in which end-users influence how a design takes shape" [40], meaning users are involved in every stage of development. As end-users are the potentials users of the product, it is important the main focus is always on them during the entire process.

To involve users in this process, a set of tools help along the way: first, define personas, the main target users of the system being developed. Personas are abstractions of real potential users who share some characteristics and needs, meaning personas are represented in a fictional individual that represents the group of real users with the same or similar characteristics [41]. Although being fictional, personas come along with a name and a photo to represent the individual. Also, personas are described in a narrative form for two purposes: first, so people see the persona as a real person and also to give a whole story about the concerns and needs of the persona regarding the product that is being developed [42]. The description of the persona is based on the type of individual: age, occupation, where the persona lives, what he/she likes to do for fun and some other things that help the persona came to life [43]. Every persona has a motivation, that defines the reason why the persona needs a new technological solution for the problems he/she is facing. By defining personas and their motivations, it becomes more clear what the audience is expecting from our product and the problems that they face, nowadays, with the existing technology. We can divide personas into three categories:

- **Primary personas**: Users who will use the system directly. They are the end-users, the main target.
- **Secondary personas**: Users who, despite not using the system, are affected by the system.
- **Served personas**: People who benefit from the usage of the system.

After defining the set of personas and their motivations, taking into account what they are expecting and the troubles that they face, scenarios are developed in order to simulate the usage of the system by a certain persona. Every UCD methodology incorporate scenarios but the main challenge is to make it more efficient [44]. A scenario is a description of a set of tasks that an user performs or wants to perform and with that, it is possible to sketch a future technology that will help users perform the tasks, It is a real vision on how a person can do something supported by technology [45]. Scenarios are concrete instances of system use that include space, time, people, and system features, while still providing not only designers but also developers and stakeholders with valid units on which to support their analyses [46]. Scenarios are supposed to tell stories on how an user could use the future platform. It is supposed to point how the main features of the system which users will use, how, when and where users will use the system.

After developing the set of personas and a considerable number of scenarios, the requirements of the system can be identified from that. It is in the requirements phase that the interests of all stakeholders of the project, like costumers, developers, users and many others, came together. If it is handled in the right way, it can lead to make both costumers and developers extremely happy and satisfied, but if it is badly worked it usually lead to a weaker quality of the product and reduces business value [47]. Scenarios, sequence of events described in a narrative way, is one of the most efficient ways to stakeholders discover requirements [48]. The main objective is to have a good set of requirements in order to be able to proceed with the design of the product, knowing that the level of risk of rework or building products that are not according to persona's will is minimum [47].

As User Centered Design methodology is an iterative process, we can change the description of the personas and scenarios whenever we want but always before finalizing the whole process, and, with that, change some or even all of our requirements. After the process is closed, we cannot go back and change to adapt to what we have. It is supposed to follow closely, and whenever we have an idea for the system make sure that idea meets the persona's will and the requirement list. If the idea is not aligned with the persona's motivation and the requirements of the system, then it is probably not a good idea for the technological solution being developed.

After that, using the framework RASA, is is possible to develop several versions of the assistant and, later, instantiate them, using RASA X, to do the first few tests, because RASA X is a platform that could run the assistant either locally or in a server, that enables the possibility of being tested by other users easily. With that, the system can be evaluated, to see if the basic functionalities are working, meaning, the assistant can be presented to testers in order to evaluate it. The first version of the assistant act as a prototype that is evaluated and using testing outcomes the assistant improves, whether by changing some intents/actions to perform some request users may have or just by adding some examples to intents so it is easier for the assistant to recognize the intent based on user's sentences. With this changes and improvements, it comes to live several improved versions of the assistant that needs to be evaluated in order to check if it is going to meet user's needs and motivations for the system.

2.5.1 Conversational Driven Development

One of the important features of RASA that allows to stick to the iterative process in developing and evaluating Conversational Assistants is the Conversational-Driven Development (CDD) methodology. CDD is a way to keep the iterative process in Conversational Assistants and it states that developing great assistants is challenging because, typically, users say unexpected things that you didn't anticipate, so the best way is to listen to your users and use those insights to improve your assistant. So, instead of trying to build the best assistant right way, it is best to develop an assistant with a small set of features and **share** your assistant with users as soon as possible. Then, ask users to talk with your assistant and, by **reviewing** conversations on a daily basis, it is possible, not only to **annotate** messages and use them as NLU training data but also to **track** when your assistant fails and measure its performance. As CDD is not a linear process, you'll circle back over and over again while developing your assistant. So, once you build an assistant with the most important features, share it with users. The way to share the assistant with users in RASA is RASA X. With RASA X is possible to generate a link so users can talk to the assistant and in the backend review all the conversations users had with the assistant: annotate the user's sentences to add to NLU training data, correcting the identified intents in case it is wrong and other functionalities.

2.6 Conclusions

It is important to note that all people with disabilities have special needs other people don't have. For PwD access to information on the accessibility of spaces if of utmost importance and with the common hierarchical menu systems they find hard to search for the specific accessibility information they want. So, it is important to propose a system that tackle the needs of PwD and make it easier to search in a system that has all the information regarding accessible tourism.

Of everything I have reviewed, what is still a challenge is the amount of information available which increases the difficulty to provide that information to people with different abilities and needs. In this regard, what seems a promising option to explore is a Conversational Assistant for Accessible Tourism. The review of the literature showed that the biggest issue for PwD regarding tourism is to access the information about accessibility items like ramps, size of the bedrooms, information in braille. Despite an hotel may have all the accessibility requirements available, if that information doesn't exist or it is really hard to find, most of the people with disabilities that needs to be sure that an hotel, in fact, does have all the access requirements they need, the most likely thing to occur is that those people don't feel comfortable to go. So, it is crucial, not only that this information indeed exists, but also that is easy to find and that is truthful. With an assistant, by providing a more natural method for requesting and providing the information, it may be possible to avoid the amounts of information of common portals and surpasses difficulties dealing with lack of technological pro-efficiency.

To develop a conversational assistant and having the needs of PwD as the major concern, it is important to do it in a user centered design methodology. So, in the next chapter, it is presented the method followed to achieve a user-centered-design approach: definition of a set of personas, set up scenarios for each persona and identify the requirements of the conversational assistant for accessible tourism.

CHAPTER 3

Personas, Scenarios and Requirements

In this chapter, the main goal is to describe the methodology followed in order to reach a system with the user's motivations in the center. For that purpose, it is important to understand the users: define personas, a representative depiction of the users, and their motivations, understand how users can deal with the system, using scenarios to describe it and, last, identify the requirements of the conversational assistant using the scenarios to do it and based it on user's interests and needs.

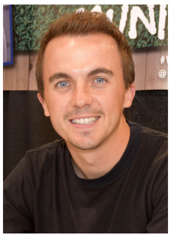
3.1 Understanding the Users

The work carried out in research project ACTION served as grounds to develop the set of personas. In the scope of the project, questionnaires were made to several people, all of whom had already taken tourist trips, with different disabilities: a hundred and twenty four people with physical disability, forty seven people with visual impairment, twenty five people with hearing impairment and a hundred and ten with intellectual disability. Questions like what type of accommodation and transports they normally use during tourist trips, with who they travel and the type of activities they normally do were made in order to better understand their difficulties and the type of trips they usually engage in. Besides that, a subject that was, also, important to know was where they search the information to plan tourist trips and the level of satisfaction with both the information and the platform used to collect the information. In a scale of 1 to 5, the average of the four groups evaluation, regarding the level of satisfaction with the information and the accessibility of the websites was a 3. To end the questionnaires, the last questions were about a possible new platform to provide information regarding accessibility: what type of accessibility items they want the platform to support and what general characteristics the platform should have such as present information in a simple language.

The information coming from the questionnaires along with the helpful guidance coming from domain experts enable the definition of a set of personas describing different types of disabilities.

3.2 Personas

The methods described above resulted in a set of Personas that represent, in a wide range, our end-users. Bellow it is described the four personas considered for this system.



José Alberto Dias is a mathematics teacher at António Damásio Secondary School, is 37 years old, lives in Lisbon and is deaf. He enjoys traveling with his family and visiting new places. In the scope of his work, he usually performs some national and international trips, but sometimes he finds difficulties in these trips because, by being deaf, he would like finding more accessible information about places to visit and if they have ways to facilitate the visit for the deaf people. Even so, he travels because it makes him happy and he enjoys meeting new people and realities.

Motivation: José wants to continue traveling for leisure and for work and for that to happen he would like to be able to search in an intuitive way, with an accessible and simple.

Inês Oliveira, 20 years old, is a student at the University of Algarve, where she lives with her parents and brother and is blind. On vacation, she usually travels with her family or friends, in order to discover new destinations and cultures. When preparing a trip, Inês wastes a lot of time searching for places to visit with her family because she doesn't find information about whether it is suitable or not for the blind and, when she does, the information is either scattered across various platforms or not coherent, which makes the whole experience more difficult. Inês also thinks that it makes her parent not risk going to visit places that Inês thinks she would love.

Motivation: Inês would like to find a system that compiles all the information necessary to book your trip and that is easy to navigate, so that you don't have to access different platforms and that this information is reliable. Inês would also like to have feedback from other people who are in the same situation as Inês about her experience.

Fernanda Almeida is a psychologist, is 45 years old, lives in Porto and is a quadriplegic. She likes to go out and visit new places. She usually travels alone, with family, or with friends. She, preferably, uses its own vehicle when traveling. Often, when planning or going on a trip, Fernanda spends her time researching places she finds interesting, but which do not have the conditions that would allow her to fully enjoy the experience, which makes Fernanda give up visiting them and makes the whole trip less rewarding and fun.

Motivation: For Fernanda, it is essential to provide detailed and specific information on the physical accessibility of different equipment and spaces, both public and private, and with the support of images.

Marta Pereira is 22 years old, is a barman in the social organization of her city, where she also practices swimming and bocce and has intellectual disabilities. She likes to travel and usually does so, especially within the scope of the association's activities. She likes to participate in the decision of the places and destinations to visit. She needs some support to plan trips that she gets from the association and his family, but she is a little anxious about not knowing what the routine of her trip is.

Motivation: Marta would like to be more participative in the planning and decision of places to visit and activities to be carried out, and she would also like to have a way to define in advance how her travel routine will be so that she could consult during it.

3.3 Scenarios

Taking into perspective the personas presented above, scenarios were created to demonstrate how a certain user could benefit by using our platform and how and where it would occur. Scenarios help clarify the flow of actions that is required to support and it provides the possibility to think about the interaction with the system before developing it. First, the scenarios were developed for an overall system to exemplify the flow of actions that it was required to support and the following scenarios illustrate that role.

In a first step, the scenarios were designed taking into account a general system and not a conversational assistant. This led to interactions not so oriented towards the conversational style that normally takes place in the presence of conversational assistants. In a second phase, when it was already certain that the assistant was the technological solution to follow, the scenarios were slightly modified to better incorporate the idea of a conversational assistant for accessible tourism. So, the next sections are divided into two parts: first the initial scenarios ideas, oriented towards a general system to give accessible information to PwD and then the scenarios for the conversational assistant. It was from the second scenarios that the requirements were extracted.

3.3.1 Initial ideas

José goes to a conference

José will have to travel to Barcelos, on business, because he will have a conference. As the conference will take place over the weekend, this will force him to sleep in Barcelos from Saturday to Sunday. José enters the platform, chooses the hotels option and then searches for Barcelos to find a list of hotels in Barcelos. The platform not only offers him a list of possibilities, but also two suggestions based on the feedback given by other people who really liked those two hotels. After seeing the two suggestions from the platform, José consults the photos of these hotels to see which one looks the most modern. He access the photos and choose Art'Otel Barcelos.

José spots a museum

On his way to the restaurant to have dinner, José noticed a museum dedicated to art on one of the streets near his hotel and thought it might be interesting to visit it the next day. After dinner, when he returned to the hotel room, he decided to get some information about the accessibility conditions of the museum. When entering the system, as he did not remember the name of the museum, he chose an option which allows him to see places of interest in the surrounding area. He found the museum that he was looking for but, unfortunately, it did not offer guided tours in Portuguese Sign Language (PSL). José was sad because it seemed like the perfect activity for the following day. Then he noticed that the platform suggested some options that might be of his interest: he spotted a theater play, the day after at 10 am, with PSL interpretation, 10 minutes away from his hotel. As José would only attend the afternoon conference on Sunday, he had the morning available to watch this play.

Inês goes to Lisbon

Inês is finishing the details of her trip to Lisbon and, to do that, she only needs to choose the hotel. Inês asks the system "I want hotels in Lisbon close to the center". The system responds "The suggestions for hotels in Lisbon are the Hotel White Lisboa and the Rossio Garden Hotel". As Inês remembered that a friend, that was also blind, told her that the Rossio Garden Hotel was a very pleasant surprise and that the entire staff were friendly and caring, she choose that one. But, just to be sure, Inês asked the system "does Rossio Garden Hotel have braille information to access bathrooms and the elevator?" to which the system replied "Yes, the hotel does".

Inês wants to visit Lisbon

Inês just finished her visit to the Jerónimo's Monastery and is excited to visit other places that are close by and that provide an audio guide. So, Inês asks the system "look for a museum that has an audio guide and that is close to my location". The system replies, "There are three possible locations in the vicinity. The most frequented is the National Museum of Archeology. Do you want another option? " As Inês does not know if she will like the museum she replies "yes, give me another option for a museum". The system answers "another suggestion is the Centro Cultural de Belém". Inês has always heard a lot about the Center but has never visited it, so she decides that it is the next tourist spot to visit.

Fernanda needs a break for vacations

Fernanda wants to go on vacation with her family to Algarve and, because of that, she opens a new conversation with the conversational agent and writes “hotels in the Algarve near the beach”. Beside this informations, the assistant also add to this research the characteristics that Fernanda indicated in her profile, such as physical accessibility and the need to provide images and measurements of the accommodation so that she can understand whether or not it is suitable for her. The system presents her with a set of options, but suggests Lagoa Hotel and Hotel Ibis, in Faro, because they meet the accessibility requirements. For the Ibis Hotel, the system adds that there is positive feedback about the existence of ramps throughout the hotel, which greatly facilitated the journey of a user who has the same characteristics as Fernanda. Fernanda navigates through the different options and notices that the two have adequate accommodation measures and that they are very close to the beach, but as the Ibis hotel presents positive feedback, she trusts this choice and starts to prepare for her trip to Faro.

Fernanda wants to visit a museum

Upon arriving at Algarve, Fernanda remembers that a friend recommended her to visit the Museum of Portimão but when consulting the system she realizes that the Museum is not very suitable given the feedback from other people who visited it under the same conditions as Fernanda. Then she opens the conversation with the conversational agent and search for “museums in Portimão”, museums that are appropriate to her conditions described in the characteristics of her profile. The system suggests the "Centro de Ciência Viva de Lagos", but it adds that there is another option that meets all the criteria. Fernanda consults the accessibility conditions and sees photos of the Center and notices that the Center is very adequate. Fernanda’s daughter is overjoyed when she realizes that they can visit the Center because she is a fan of science. At the end of the visit, she is notified by the system that asks if she enjoyed the visit and if the Center was suited to her characteristics. Fernanda, as she values other people’s feedback, gives her opinion about the Center.

Marta and her collegues choose an hotel to stay

A monthly trip is planned in the institution that Marta attends. The time has come to plan the trip for June. An employee, Amílcar, access the system to search accommodations that are accessible and that are able to accept groups of twenty people. After the first selection, Amílcar goes to the common room where Marta and the other users of the institution meet to choose which hotel to stay in. The group is excited to be able to participate in the process and Amílcar starts by projecting television images of hotels so that Marta and her colleagues can give their opinions. The whole group goes through the different photos and reacts to each one of the photos, indicating in the system whether they like it or not. In the end, Amílcar chooses the hotel that obtained the highest number of likes of the institution users.

Marta and her activities

In order to prepare for the trip, the technicians and Marta talk about the type of activities she would like to do during the trip. Marta is usually a little anxious about these outings, because she is used to following a routine in her day-to-day basis and all these new things unpredictable inherent to a trip make her a little anxious, so she likes to participate more in the process so she can understand how it will be and where it will be. Thus, while the caregiver sees possible activities on his device, he shows Marta on television images of the activities he sees for her to comment on. Marta doesn't like the first activity presented, because it involves heights and Marta has a phobia of heights. The second activity seems to be more interesting and Marta reacts, on your device, with a "like", when the activity seems to be interesting, as Amilcar introduces you to various activities.

3.3.2 Scenarios for an Accessible Tourism Conversational assistant

Starting from the initial idea, scenarios were expanded taking into consideration the conversational assistant. Bellow there are described the scenarios with those slight changes with the requirements at red with the priority level: P0 for absolute top priority to integrate in the first version of the conversational assistant and P2 for least priority requirements.

José goes to a conference

José will have to travel to Barcelos, on business, because he will have a conference. As the conference will take place over the weekend, this will force him to sleep in Barcelos from Saturday to Sunday. José enters the platform [REQ: P0 – Assistant available in browsers], and types "hotels in Barcelos". [REQ: P0 – Allow written input] [REQ: P0 – Answer questions on Hotels in a place] The platform offers him one suggestion. After that, José consults the photos of this hotel, asking "Can i see the photos of that hotel?"

[REQ: P2 – Be able to perform NLU processing of questions regarding photos of hotel or museums] to see if it looks modern. He sees the photos [REQ: P2 – Provide photos of the hotel] and choose the suggestion Art'Otel Barcelos.

José spots a museum

On his way to the restaurant to have dinner, José noticed a museum dedicated to art on one of the streets near his hotel and thought it might be interesting to visit it the next day. After dinner, when he returned to the hotel room, he decided to get some information about the accessibility conditions of the museum [REQ: P0 – Provide accessibility of the museums]. When entering the system, as he did not remember the name of the museum, he search for museums in Barcelos. He found the museum that he was looking for but, unfortunately, it did not offer guided tours in Portuguese Sign Language (PSL). José was sad because it seemed like the perfect activity for the following day. Then he searched for locals of interest in Barcelos [REQ: P0 – Search for locals of interest/events in a location] and received as response a theater play, the day after at 10 am, with PSL interpretation, 10 minutes away from his hotel. As José would only attend the afternoon conference on Sunday, he had the morning available to watch this play.

Inês goes to Lisbon

Inês is finishing the details of her trip to Lisbon and, to do that, she only needs to choose the hotel. Inês asks the system [REQ: P0 – Assistant available in browsers] [REQ: P1 – Allow voice input] “I want to search for hotels"

[REQ: P0 – Be able to perform NLU processing of questions regarding hotels] which the system replies "where do you want to search hotels?". Inês responds "Lisbon" [REQ: P0 – Be able to do NLU processing of possible locations for hotels, museums or locals of interest]. The system responds "The suggestion for hotel in Lisbon is the Lisboa Pessoa Hotel". "[REQ: P0 – Answer questions on Hotels in a place]. As Inês remembered that a friend, that was also blind, told her that the Lisboa Pessoa Hotel was a very pleasant surprise and that the entire staff were friendly and caring, she chose that one. But, just to be sure, Inês asked the system “which facilities there are in that hotel?”

[REQ: P0 – Be able to perform NLU processing of questions regarding generic accessibility information on a specific slot: ask for accessibility information of an hotel, museum or local of interest] to which the system replied “Hotel Pessoa Lisboa Hotel has the following accessibility conditions: Braille information of the restaurant’s menu, professionals with sign language skills, restaurant with adapted tables for wheelchair access” . [REQ: P0 – Answer questions about accessibility in general of an hotel, museum or location]

Inês wants to visit Lisbon

Inês just finished her visit to the Jerónimo’s Monastery and is excited to visit other places that are close by and that provide an audio guide. So, Inês asks the system “look for a museum in Barcelos”.

[REQ: P0 – Be able to perform NLU processing of questions regarding Museums] The system replies, “There are three possible locations in the vicinity. The most frequented is the National Museum of Archeology.

[REQ: P1 – Search information to support answers for museums in a location] As Inês does not know if she will like the museum she replies "I want another option.”

[REQ: P0 – NLU processing of questions requiring other option of the same type: hotel, museum or location] The system answers “another suggestion is the Centro Cultural de Belém”.

[REQ: P0 – Answer questions of other hotels, museums or locals in the same location] Inês has always heard a lot about the Center but has never visited it, so she decides that it is the next tourist spot to visit.

Fernanda needs a break for vacations

Fernanda wants to go on vacation with her family to Algarve and, because of that, she opens a new conversation with the conversational agent [REQ: P0 – Assistant available in browsers] and writes “hotels in the Algarve near the beach”[REQ: P0 – Search for hotels in a place]. The system tells her there are a set of options, but suggests Lagoa Hotel and Hotel Ibis, in Faro, because they meet the accessibility requirements. Fernanda asks for feedback on Ibis Hotel [REQ: P2 – Ask for feedback of an hotel] and the system response is that there is positive feedback about the "existence of ramps throughout the hotel, which greatly facilitated the journey" of an user who has the same characteristics as Fernanda. Fernanda also asks for accommodation measures on Ibis Hotel [REQ: P1 – Ask for accessibility of an hotel] and the response satisfies her needs.

Fernanda wants to visit a museum

Upon arriving at Algarve, Fernanda remembers that a friend recommended her to visit the Museum of Portimão but when consulting the system she realizes that the Museum is not very suitable given the feedback from other people who visited it under the same conditions as Fernanda [REQ: P2 – Ask for feedback of a museum] . Then she opens the conversation with the conversational agent [REQ: P0 – Assistant available in browsers] and search for “museums in Portimão” [REQ: P0 – Search for museums in a place]. The system suggests the "Centro de Ciência Viva de Lagos", but it adds that there are other options. Fernanda consults the accessibility conditions [REQ: P1 – Provide accessibility of a museum] and sees photos of the Center [REQ: P2 – Provide photos of the museum] and notices that the Center is very adequate. Fernanda’s daughter is overjoyed when she realizes that they can visit the Center because she is a fan of science. At the end of the visit, she is notified by the system that asks if she enjoyed the visit and if the Center was suited to her characteristics. Fernanda, as she values other people’s feedback, gives her opinion about the Center [REQ: P2 – Possibility of providing feedback].

3.4 Requirements

Starting from the scenarios for the conversational assistant, the requirements were identified with different levels of priority: **P0** for the requirements that were mandatory for the first version of the conversational assistant, **P1** to the ones who should be implemented later and **P2** for the ones that would only be implemented if there were time.

1. P0: Assistant available in browsers
2. P0: Allow written input
3. P1: Allow input by voice
4. P0: Be able to perform NLU processing of greetings from users
5. P0: Be able to perform NLU processing of questions regarding hotels
6. P0: Be able to perform NLU processing of questions regarding locals of interest
7. P0: Be able to perform NLU processing of questions regarding Museums
8. P0: Be able to perform NLU processing of questions regarding generic accessibility information on a specific slot: ask for accessibility information of an hotel, museum or local of interest
9. P0: NLU processing of questions requiring other option of the same type: hotel, museum or location
10. P0: Be able to do NLU processing of locations to search for hotels, museums or locals of interest there
11. P1: NLU processing of questions regarding more specific accessibility: ask for specific information like access to wheelchair
12. P2: Be able to perform NLU processing of questions regarding photos of hotel or museums
13. P2: Be able to perform NLU processing of questions regarding feedback of other users
14. P0: Greet the users
15. P0: Answer questions on Hotels in a place
16. P0: Answer questions for locals of interest/events in a location
17. P0: Answer questions on museums in a place
18. P0: Answer questions of other option for hotels, museums or locals in the same location
19. P0: Answer questions about accessibility in general of an hotel, museum or location
20. P1: Search information to support answers for locals of interest/events in a location
21. P1: Search information to support answers for hotels in a location
22. P1: Search information to support answers for museums in a location
23. P1: Answer questions on more specific items of accessibility of an hotel, museum or location
24. P2: Answer questions of feedback of an hotel, museum or location
25. P2: Provide photos of the hotel, museum or local of interest

3.5 Conclusions

This chapter describes the methodology followed to ensure that the system's requirements is based on the user's needs and motivations. For that goal to be achieved it was important to define the set of personas to consider for the system and elaborate a number of scenarios for each persona. Following a first set of scenarios illustrating the actions performed by the Personas, a second set of scenarios was proposed considering conversational features and, from these, a set of requirements were identified. These will serve as grounds for proposing an assistant for Accessible Tourism.

CHAPTER 4

Overview of Conversational Assistant Development in RASA 2.0

As briefly explained in chapter 2, the RASA 2.0 framework provides a wide range of functionality for the development, test and refinement of Conversational Assistants. As a first stage of the work carried out, the framework was explored to get familiarized with the numerous features of RASA. In short, this chapter is a summary of the information that is most relevant for the work to be carried out, distilling the vast documentation of RASA. In this sense it can be understood as the stepping stones to use RASA to develop a conversational assistant.

4.1 Overall concepts regarding developing a RASA Conversational Assistant

To develop a RASA Conversational Assistant it is important to address several concepts that are the central piece of the assistant, like, for example, intents, actions, slots and policies. Below it is addressed every concept.

NLU intents - define several intents and add user utterances examples to each intent.

- **Define intents**: taking into account the different functionalities of the assistant and the dialog acts define the different intents that you want your assistant to recognize from user input.
- **Add examples**: For each intent add some utterances to help identify the intent based on user's input.

Actions - there are four types of actions the assistant can provide to the user: **responses**, **custom actions**, **forms** and **default actions**. Responses are normally only text, images or similar. Custom actions can run any code you want, like API calls. Forms are normally slot filling actions, where you keep asking the user to fill slots you need in order to do something, like booking a restaurant. Default actions you don't need to do anything.

Define actions: taking into account the dialog acts define the several responses the assistant should perform. Unlike intents, in responses you don't need to add several examples for each response, one is enough. If you want your assistant to have different responses for the same response you can add more than one example and whenever the response is triggered, it is chosen one randomly.

Dialog Management - Use stories and rules to train the dialog management.

- **Stories** are a sequence of steps with both intents perceived from user's input and the actions performed by the assistant and can generalize to unseen conversation paths.
- **Rules** have the same steps as stories but can't generalize for unseen conversation paths. You can apply conditions in which the rule applies.

Dialogue Policies - Choose between the several policies available in RASA.

There is one rule-based policy, called RulePolicy and is based on file `rules.yml` and machine-learning policies (TEDPolicy, MemoizationPolicy, etc) that you can use in your assistant. You can use more than one policy but, because each policy returns a next action with a certain confidence you need to prioritize policies so if 2 policies return a next action with the same confidence the assistant chooses one.

Train - Train the assistant in command line

After define all that, you can train the assistant in the command line. That creates a model that is placed inside models directory and you can test the assistant in the terminal if you have both commands **`rasa shell`** and **`rasa run actions`** running in your terminal. Every time there is a change in either `actions.py` file run **`rasa run actions`** again and if there is any change in the model run **`rasa train`** to train the new model.

4.2 Different stages in developing a RASA Conversational Assistant

This sections details the different steps that you need to take in order to develop an assistant in the RASA framework, its details and, after that, a real example code for each one of the different steps.

Steps:

1. NLU intents
2. Actions: Responses, Custom Actions and Forms
3. Slots - Passing information
4. Dialog Manager
5. Dialogue Policies
6. Train and talk to the assistant

step 1: NLU intents

- Analysis of the problem: dialog acts are crucial to have a more clear sight of what the intents may be. After analysing a possible conversation with the assistant we can extract the main features, in other words, the intents of the user and the responses of the conversational agent.
- Define intents: they are the main goals of the assistant, the purpose of having a conversation, what the user wants to know (for example, if it is meant to look for restaurants in a certain city, probably the intents will be greeting and finding_restaurant)
- Provide possible examples of sentences for those intents (example: for intent finding_restaurant: `Help me search a restaurant / Find me a restaurant / Which restaurants are in Aveiro?)`
- Entities: You can add entities that are structured pieces of information in a user's message. For entity location, an examples is: `Which restaurants are in [Aveiro](location)?`
- Steps to take: intents are placed in data/nlu directory and in file domain.yml under intents key. In file domain it is only mandatory to have intent's name and not all text examples of that intent.
- Real example: code

```
nlu:
- intent: greeting
  examples: |
    - Hi
    - Hey!
    - Hello
    - Good day
    - Good morning

- intent: get_horoscope
  examples: |
    - I want to get the horoscope
    - Can you send me the horoscope?
    - I want to see my horoscope?

- intent: inform_horoscope
  examples: |
      -lion
      -peixes

- intent: subscription
  examples: |
    - Please subscribe me
    - Can i subscribe?
```

step 2: Actions: Responses, Custom Actions, Forms

Responses: Responses are the phrases the assistant will say based on what was said by the user. Unlike the examples above of the intents (possible input by the user), the output is always one of the examples we write on responses, so if we want the assistant to have different greetings or different phrases for the same response we need to add those examples here.

- Define responses: It is also based on dialog acts as intents: the main difference is that, in responses, we need to add several examples for the same response if we want our assistant to have different ways of greeting the user or different ways of saying the "same thing". Responses are the phrases the assistant send without having to perform any service: it is only a response. API calls or access to databases are not categorize as responses but as actions. Responses name should start with "utter_". For example, for intent "greeting", we should have a response name "utter-_greeting".
- Add examples for responses: Here, we should diversify the responses, so the user does not receive the same message everytime for the same intent.
- Steps to take: Under the responses key in your domain file.
- Real example: Code

```
responses:
   utter_greet:
       - text: |
           Hello! How are you doing today?
       - text: |
           Hi!
   utter_ask_horoscope_sign:
       - text: |
           What is your horoscope sign?
   utter_subscribe:
       - text: |
       Do you want to subscribe for daily updates?
    utter_ask_dob:
        -text: |
        What is your DOB in DD-MM format?
```

Actions - Everytime a response requires access to database, invoke services or API calls, actions are the solution.

- Define actions: Everytime a response needs something more than plain text, for example, if it has to run code, actions are the solution.
- How to do it: After naming the several actions in file domain.yml under actions key, in the file actions.py it is mandatory to add a class for each action and perform the code that does what we want that action to do.

- Steps to take: File actions.py is placed inside the folder actions on root directory. We also need to define actions in the domain file under actions key. After that, in a terminal run "rasa run actions" so you could talk to the assistant.
- Real example: Code

```
from __future__ import absolute_import
from __future__ import division
from __future__ import print_function
from __future__ import unicode_literals

import requests
from rasa_core_sdk import Action
from rasa_core_sdk.events import SlotSet

class GetTodaysHoroscope(Action):

    def name(self):
        return "get_todays_horoscope"

    def run(self, dispatcher, tracker, domain):
        # type: (Dispatcher, DialogueStateTracker, Domain) -> ...
            List[Event]

        user_horoscope_sign = tracker.get_slot('horoscope_sign')
        base_url = ...
            "http://horoscope-api.herokuapp.com/horoscope/{day}/{sign}"
        url = base_url.format(**{'day': "today", 'sign': ...
            user_horoscope_sign})
        #http://horoscope-api.herokuapp.com/horoscope/today/capricorn
        res = requests.get(url)
        todays_horoscope = res.json()['horoscope']
        response = "Your today's ...
            horoscope:\n{}".format(todays_horoscope)

        dispatcher.utter_message(response)
        return [SlotSet("horoscope_sign", user_horoscope_sign)]


class SubscribeUser(Action):
    def name(self):
        return "subscribe_user"

    def run(self, dispatcher, tracker, domain):
        # type: (Dispatcher, DialogueStateTracker, Domain) -> ...
            List[Event]

        subscribe = tracker.get_slot('subscribe')
        if subscribe == "True":
            response = "You're successfully subscribed"
        if subscribe == "False":
            response = "You're successfully unsubscribed"

        dispatcher.utter_message(response)
        return [SlotSet("subscribe", subscribe)]
```

Forms - Everytime it is required to fill a number of slots, like when booking a restaurant, forms are the way to do it.

- Define forms: In file domain.yml under forms key define the forms.
- Real example:

```
forms:
  restaurant_form:
    required_slots:
        cuisine:
          - type: from_entity
            entity: cuisine
        num_people:
          - type: from_entity
            entity: number
```

step 3: Slots - When you need to store some type of information said by user, like a location or the sign of the user in the case of an horoscope assistant you need to use slots.

- Steps to take: It is mandatory to define slots in domain file under slots key and the type of slot: text, Boolean or other. After that, you can set slots in file actions.py with [SlotSet("slot", variable to store in slot)]. It is also mandatory, in stories.yml file, whenever a action needs to save the slot, after the action name, it is used `"slot_was_set"` key and an example of a possible value of the slot.
- Real example:
  1. In file `actions.py`: return [SlotSet("horoscope", horoscope_sign)]
  2. In file `domain.yml`:

```
slot:
 horoscope:
  type: text
```

step 4: Dialog Manager
Stories

- Define stories: stories are the way the system knows what to say next. It consists in a series of intents and responses in a conversation style. That is, it is a sequence of one phrase written by the user and the response the system must perform. The story has to make sense.

- Steps to take: Stories are defined under the directory data in a yaml file named stories.
- Real example: Code

```
stories:
- story: greet and sign
  steps:
   - intent: greeting
   - action: utter_greet
   - intent: get_horoscope
   - action: utter_ask_horoscope_sign
   - intent: inform_horoscope
   - action: get_todays_horoscope
   - slot_was_set:
      - horoscope_sign: lion
   - action: utter_subscribe

- story: greet and subscribe
  steps:
    - intent: greeting
    - action: utter_greet
    - intent: subscription
    - action: subscribe_user
    - slot_was_set:
       - subscribe: true
```

Rules

- Rules are great for small and specific conversation patterns. They can´t generalize for unseen conversation patterns.
- Steps to take: It is stored under data directory in a file named rules.yml. It consists in a number of steps, just like stories, with an intent and the respective response. Also, can include conditions which that same rule applies.
- Real example:

```
rules:
- rule: Say `hello` whenever the user sends a message with intent ...
    `greet`
  steps:
  - intent: greet
  - action: utter_greetty: number
```

step 5: Dialogue Policies

- Define policies: You can choose between all dialogue policies available, even choose more than one policy for the same assistant.

- Steps to take: Define policies in the config.yml file in root directory using the policy name.
- Real example: code

```
language: pt
policies:
  - name: MemoizationPolicy
  - name: TEDPolicy
    max_history: 5
    epochs: 200
  - name: RulePolicy
```

step 6: Train and talk to the assistant

- After train the assistant (use command **rasa train** to train the new model), the model is stored in models folder on root directory. Whenever you change your model (intents, responses, stories, etc) you need to run **rasa train** again to update to the new model.
- To talk with the assistant, run "rasa shell" in one terminal and if you have actions in file actions.py you also have to run command "run rasa actions" in another terminal.
- With the command "rasa shell –debug" you can see extra information like, for example, the percentage of confidence in a certain intent and you can have a clear sight of the background, that is, everything that is done in order to find the perfect response to what the user said.

CHAPTER 5

Conversational Assistant Development and Deployment

The assistant was developed in a CDD methodology, through iterative use by members of the research group at IEETA and in two occasions a more formal process was conducted with domain experts. The development and deployment of the Conversational Assistant for Accessible Tourism included a number of steps:

1. **Development of the First Version of the Conversational Assistant**: using RASA, the first version was made and, using the command line, the first few tests to the assistant were performed locally to make sure the assistant was behaving the way it was supposed to before going the next step.
2. **Deployment in RASA X and First Use by Domain Experts**: RASA X to do the deployment of the Conversational Assistant so other people could test it and the conversations could be continuously analyzed in order to improve the assistant.
3. **Iterative Improvement of the Conversational Assistant**: based on the evaluation of the first version plus the requirements that weren't taking into account for the development of the first version of the Conversational Assistant.
4. **Use of the improved version by domain experts**: the same approach as in the first use but extending it for a larger number of people.

5.1 Development of the First version of the Conversational Assistant

After developing the personas and scenarios for the system, it was established that a conversational assistant could be a good way to improve accessible tourism for people with disabilities.

Considering the several motivations of our 4 personas and also, taking into account the most searched items when looking for accessibility information, a number of items (requirements) for the conversational assistant were prioritized for the first sketch of the conversational assistant:

5.1.1 Selected requirements

The set of requirements evaluated with priority equals to P0 were selected to be implemented in the first version of the conversational assistant:

1. Assistant available in browsers
2. Allow written input
3. Be able to perform NLU processing of greetings
4. Be able to perform NLU processing of questions regarding hotels
5. Be able to perform NLU processing of questions regarding Locals of Interest
6. Be able to perform NLU processing of questions regarding Museums
7. Be able to perform NLU processing of questions regarding generic accessibility information on a specific slot: ask for accessibility information of an hotel, museum or local of interest
8. NLU processing of questions requiring other option of the same type: hotel, museum or location
9. Be able to do NLU processing of locations to search for hotels, museums or locals of interest there
10. Do opening greetings of the dialogue
11. Answer questions on Hotels in a place
12. Answer questions for locals of interest/events in a location
13. Answer questions on museums in a place
14. Answer questions of other option for hotels, museums or locals in the same location
15. Answer questions about accessibility in general of an hotel, museum or location

5.1.2 Step 1: Intents

Bearing in mind the dialog acts and the requirements selected, the following intents were considered for the first version:

- `greeting`
- `request_hotel / inform_interest_hotel`
- `request_museum / inform_interest_museum`
- `request locals_of_interest`
- `inform_interest_local_of_interest`
- `inform_location`
- `request_accessibility`
- `ask_for_other_option`

Four or five text examples were added to each intent to make it easier to recognize the intent.

5.1.3 Step 2: Responses and Actions

The only 2 responses considered for the first version were the following:

- `utter_greet`: to greet the user
- `utter_ask_location`: After a request of an hotel, museum or local of interest so the assistant knows where to search.

Considering the above NLU intents, the following actions were performed:

- `save_location`
- `give_hotel`
- `give_museum`
- `give_locals_of_interest`
- `inform_accessibility`
- `give_more_information`

The action `save_location` was required so we could save the location in a slot and, with that, using action `ask_for_other_option`, the user could ask for other example of hotel, museum or local of interest in the same location.

5.1.4 Step 3: Slots

The information that the assistant needs to keep track are location, hotels, museums and locals of interest. So, 4 slots were created in order to save those names and, every time it was necessary, access the slots in actions.

- `location`
- `hotel`
- `museu`
- `locais`

5.1.5 Step 4: Stories and rules

Stories are the way the assistant knows what to say next based on the context of the conversation. They consist in a number of steps, intents and actions, that make up a story. In the examples bellow, two stories that represent a typical flow of the conversation, where the user, in the first story ask for an hotel in a location and the accessibility of that hotel, and in the second ask for a museum, but doesn't provide a location, so the assistant asks for a location, saves it and then gives a museum in that location and, after, the accessibility of the museum. More stories were added, to give locals of interest and other options in the same location, for example.

```
version: "2.0"
stories:
- story: Hotel e acessibilidade do mesmo
  steps:
  - intent: greeting
  - action: utter_greet
  - intent: request_hotel
  - action: save_location
  - slot_was_set:
    - location: Porto
  - action: inform_hotel_name
  - slot_was_set:
    - hotel: UmHotelnoPorto
  - intent: request_acessibility
  - action: inform_acessibility

- story: Museu e acessibilidade do mesmo
  steps:
  - intent: greeting
  - action: utter_greet
  - intent: inform_interest_in_museum
  - action: utter_ask_location
  - intent: inform_location
  - action: save_location
  - slot_was_set:
    - location: Lisboa
  - action: give_museums
  - slot_was_set:
    - museu: MuseuLisboa
  - intent: request_acessibility
  - action: inform_acessibility
```

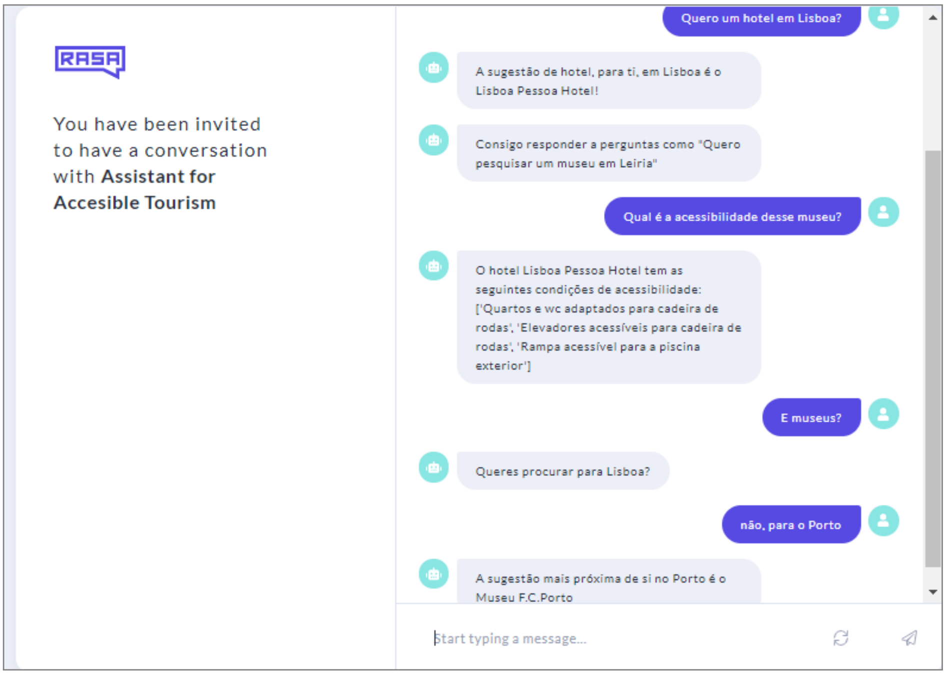
No **rules** were added for the first version of the assistant.

5.2 Deployment in RASA X and First Use by Domain Experts

The development of a conversational assistant to the point that it is able to provide a good performance, overall, takes a lot of time and resources. Beyond supporting a set of specific features, training it to understand the relevant intents and entities requires a rich set of data that cannot often be gathered in reasonable time. In our case, we adopt a conversational-driven development (CDD) approach where the assistant is deployed with a minimal set of capabilities and it is the data from its interaction with users that is considered for its continuous refinement.

5.2.1 Methods

After a first instantiation of the assistant considering the requirements and intents described above, we adopted RASA X for the deployment of the assistant. RASA X is a platform that integrates with RASA and provides two important features supporting the assistant's development and refinement: (1) the ability to generate links for sharing access to the assistant in a web interface (see figure 5.2); and (2) a back-end where all the conversations with the assistant are made available and can be explored and annotated by the developer, for instance to add user sentences to the pool of data to train the NLU, correcting the recognized intents, correcting the assistant's responses, or adding content to entities.



You have been invited to have a conversation with **Assistant for Accesible Tourism**

Quero um hotel em Lisboa?

A sugestão de hotel, para ti, em Lisboa é o Lisboa Pessoa Hotel!

Consigo responder a perguntas como "Quero pesquisar um museu em Leiria"

Qual é a acessibilidade desse museu?

O hotel Lisboa Pessoa Hotel tem as seguintes condições de acessibilidade: ['Quartos e wc adaptados para cadeira de rodas', 'Elevadores acessíveis para cadeira de rodas', 'Rampa acessível para a piscina exterior']

E museus?

Queres procurar para Lisboa?

não, para o Porto

A sugestão mais próxima de si no Porto é o Museu F.C.Porto

Figure 5.1: Example of the RASA X interface made available for users to interact with the assistant.

The next step was to make available to a small number of researchers in ACTION team the first version of the assistant for first interactions. It is important to highlight that, at this point, the purpose is not to understand if the assistant is useful or acceptable, but to get as much information as possible about how it could typically be used to evolve it accordingly.

To this end, two domain experts were informed about the overall context and motivations considered for developing the assistant, the extent of the features currently supported (e.g., The assistant can answer questions about hotels) and were provided with a list of suggestions of actions that they could perform as a starting point for interacting with the assistant. The participants were also instructed to use natural speech and, after testing the suggested action, to move into obtaining other information judged pertinent for PwD. All these information were sent in a pdf for each domain expert that can be found in the attachments. Whenever the assistant did not provide a meaningful answer the participants were requested to tell the assistant what kind of answer they were expecting (e.g., "Assistant, you should have said..").

To take the most out of the participants' inputs, the conversations were continuously analyzed, typically everyday so both participants could see improvements and didn't encounter the same problem over and over again, to understand the behavior of the assistant and in what ways should the assistant improve in order to fulfill the questions/intents provided by the testers. This resulted in novel data being annotated or corrected, and new intents and actions added resulting in novel models that replaced the initial model provided to the participants. The main purpose of this approach was to make noticeable, to the participants, that the assistant was actually improving, with their participation, thus motivating further interactions.

5.2.2 Testing Outcomes and Discussion

Right from the first conversations, users were trying to end the conversation and the assistant could not understand it, so it was required the assistant could understand both greeting and goodbye phrases in order to tackle that. Also, questions about the location of places (e.g., hotels, museums) were asked several times by evaluators because it could be important for some users to know the location of a certain hotel. Because most errors identifying the intent happened when the confidence in that intent was lower than 20% it was required to add some intent to deal with it, so every time the assistant couldn't find a confidence higher than 20% or 15% for any intent the assistant would ask to rephrase instead of going for the intent with higher confidence, because sometimes it was near 0%. Overall, the evaluation of the first version of the assistant brought some changes and improvements, which was exactly what we wanted when we followed a CDD methodology for the conversational assistant. Beyond the requirements that weren't taking into account for the first version of the assistant, there were some improvements to make in the assistant based on the evaluation which led to a much improved conversational assistant

While the assistant, at this time, was still not able to provide information about restaurants or other accessibility structures, such as information in braille, the purpose of introducing the ability to identify these intents and accessibility types is to provide the users with answers that show the assistant understood what was asked from it - particularly relevant if that is a topic relevant for the domain of application -, but information is not available, at this time. So, instead of answering, e.g., "I do not understand what you want. Please rephrase." to everything that is out of scope of its current capabilities, the assistant could start answering "I am sorry, but I still do not have information about restaurants".

5.3 Iterative Improvement of the Conversational Assistant

Based on the results of the evaluation of the first version and taking into account the initial requirements extracted from the scenarios, the requirements considered for the improved version of the Conversational Assistant were:

1. The ones that come from the evaluation, that is, intents that domain experts thought that were required:

- NLU processing of phrases regarding the end of the conversation
- End the conversation with polite messages
- NLU processing of fallback: whenever the confidence of the intent was lower than 20%, fallback intent was called
- Add a fallback sentence to ask the user to rephrase
- NLU processing of questions regarding the location of hotels
- NLU processing of questions regarding the location of museums
- NLU processing of questions regarding the location of locals of interest
- Search information to support answers for the location of an hotel
- Search information to support answers for the location of a museum

- Search information to support answers for the location of a local of interest
- NLU processing of questions regarding restaurants in a location
- Search information to support answers for restaurants in a location

2. From the initial list of requirements that were not considered for the first version of the assistant:

- Search information to support answers for locals of interest/events in a location
- Search information to support answers for hotels in a location
- Search information to support answers for museums in a location
- Answer questions on more specific items of accessibility of an hotel, museum or location
- NLU processing of questions regarding more specific accessibility: ask for specific information like ramps

The following sections describe the work carried out to attend to the new list of requirements:

5.3.1 Step 1: Intents

The intents `inform_interest_hotel`, `inform_interest_museum` as well as the same intent for the local of interest `inform_interest_locals_of_interest` were deleted because they were too similar with the corresponding intents `request_*` and that normally lead to some mistakes identifying the correct intent. Instead, in actions `give_hotel`, `give_museum` and `give_locals_of_interest` if there was no location a message was send asking for the location, basically doing the same that the response `ask_for_location` did. Bearing in mind the dialog acts and the requirements selected, the following intents were considered for the improved version:

- `greeting`
- `request_hotel`
- `request_museum`
- `request_locals_of_interest`
- `inform_location`
- `request_accessibility`
- `inform_interest_accessibility`
- `ask_for_alternatives`
- `request_accessibility_type_hotel`
- `request_accessibility_type_museum`
- `request_accessibility_type_local`
- `request_location_hotel`
- `request_location_museum`
- `request_location_local`
- `request_hotel_with_access`
- `request_restaurant`

Some examples were added to the new intents and to the old intents some examples used by the people who evaluate the first version were added as well.

5.3.2 Step 2: Responses and Actions

The responses `utter_ask_location` was deleted, as explained above and the response to say goodbye and to rephrase was added. So, this version contemplate three responses.

- `utter_greet`: to greet the user
- `utter_goodbye`: to say goodbye to users
- `utter_ask_rephrase`: ask to rephrase if the confidence of the intent was lower than 20%

Some actions were changed to incorporate the restaurant option. It was possible to ask for other restaurant in the same location and ask for the accessibility of that restaurant so in the corresponding actions it was required to define that for restaurants as well.

Considering the above NLU intents, the following actions were performed:

- `save_location`
- `give_hotel`
- `give_museum`
- `give_locals_of_interest`
- `inform_accessibility`: added accessibility for restaurants
- `give_more_information`: added more information for restaurants
- `unboarding`: to give an idea of what the assistant could do
- `give_specific_accessibility`: to respond to the new intent `request_accessibility_type_*` depending on the type of accessibility the user wants to know (ramps, access to wheelchair) and where (hotel, museum, local, restaurant)
- `inform_location`: to give the location of a certain hotel, museum, local or restaurant
- `give_restaurant`: the same that existed for hotels, museums and locals of interest
- `give_hotel_with_access`: to give an hotel with a certain accessibility, for example, an hotel in Porto with ramps.

5.3.3 Step 3: Slots

The same slots: location, hotel, museum and locals that were in the first version continue to exist in this version. And it was added the slots `restaurant` and `accessibility_type`. The second one to keep track of the type of accessibility the user wants to know, for example in `request_hotel_with_access` it is required to know the accessibility type of the request.

5.3.4 Step 4: Stories and rules

The new list of intents and actions were also added in stories so the assistant know what to say next after identifying the correct intent. Overall, new stories were required for intents such as `request_hotel_with_access`, `request_location_hotel` among others. Below are described some of the new stories added, such as asking for the location of hotel, museum and local of interest in the first story; asking for a restaurant in the second one along with the request for an hotel and a museum; in the third story is the new way of asking for an hotel, instead of intent `inform_interest_in_hotel`, every request for an hotel, with or without location is defined as **request_hotel** and is in the action **give_hotel** that, depending on whether there is a location or not, it goes to inform an hotel for that location or asks the user to provide a location for the search. The last story impersonate the request for an hotel with a certain type of accessibility, such as an hotel with access for wheelchair. The stories list was the following:

```
version: "2.0"
stories:
- story: location of things
  steps:
  - intent: request_location_hotel
  - action: give_location
  - intent: request_location_museum
  - action: give_location
  - intent: request_location_local
  - action: give_location
- story: Hotel+restaurant+museu
  steps:
  - intent: request_hotel
    entities:
    - LOC: Porto
  - slot_was_set:
    - LOC: Porto
  - action: give_hotel
  - slot_was_set:
    - hotel: Tivoli
  - action: unboarding
  - intent: request_restaurant
    entities:
    - LOC: Aveiro
  - slot_was_set:
    - LOC: Aveiro
  - action: give_restaurant
  - slot_was_set:
    - restaurante: Pizzarte
  - intent: request_museum
    entities:
    - LOC: Porto
  - slot_was_set:
    - LOC: Porto
  - action: give_museums
  - slot_was_set:
    - museu: Museu do Carro
  - intent: goodbye
  - action: utter_goodbye
- story: New way to ask for an hotel
  steps:
  - intent: request_hotel
  - action: give_hotel
  - intent: inform_location
  - action: save_location
  - slot_was_set:
    - LOC: Barcelos
  - entities:
    - LOC: Barcelos
  - action: give_hotel
  - slot_was_set:
    - hotel: In Barcelos Hostel and Guest House
  - action: unboarding
  - intent: request_acessibility_type_hotel
  - action: give_specific_acessibility
  - intent: request_locals_of_interest
  - action: give_localsOfInterest
  - intent: inform_location
  - action: save_location
  - slot_was_set:
    - LOC: Aveiro
  - action: give_localsOfInterest
  - slot_was_set:
    - locais: Sé
  - action: unboarding
- story: Hotel com acesso para cadeira de rodas+restaurante
  steps:
  - intent: request_hotel_with_access
  - action: give_hotel_with_access
  - intent: request_restaurant
    entities:
    - LOC: Leiria
  - slot_was_set:
    - LOC: Leiria
  - action: give_restaurant
  - slot_was_set:
    - restaurante: ArquiCafé
```

A rule was added to deal with the `nlu fallback`. Whenever the confidence was lower than 20% the rule fallback was triggered to ask the user to rephrase. The new rule added was the following:

```
rules:
- rule: Implementation of the NLUFallback
  steps:
  - intent: nlu_fallback
  - action: utter_ask_rephrase
```

5.3.5 Entity extraction using spaCy

For the first version of the assistant and to have a quick solution that enabled a fast deployment, the assistant only recognized a small set of elements for each entity e.g. for cities it only recognize Aveiro, Porto, Lisboa, Coimbra, Faro, Barcelos, Braga and Leiria. This was naturally one point that needed improvement and demanded a more general solution. In this regard, the literature describes several solutions that can be adopted for entity recognition such as **Duckling**, **NER_CRF** and **spaCy**. Considering these alternatives and how they can be integrated with RASA, we adopted spaCy which is a free open-source library for advanced Natural Language Processing in Python. To use it in RASA, spaCy needed to be added to the RASA pipeline. Below it is shown the configuration for the RASA pipeline, including spaCy as an entity recognizer.

```
language: pt
pipeline:
- name: SpacyNLP
  model: pt_core_news_md
```

In order to adapt to the inclusion of spaCy, that detects location entities to a slot named LOC, the slot location was renamed to LOC so RASA could fill this slot automatically.

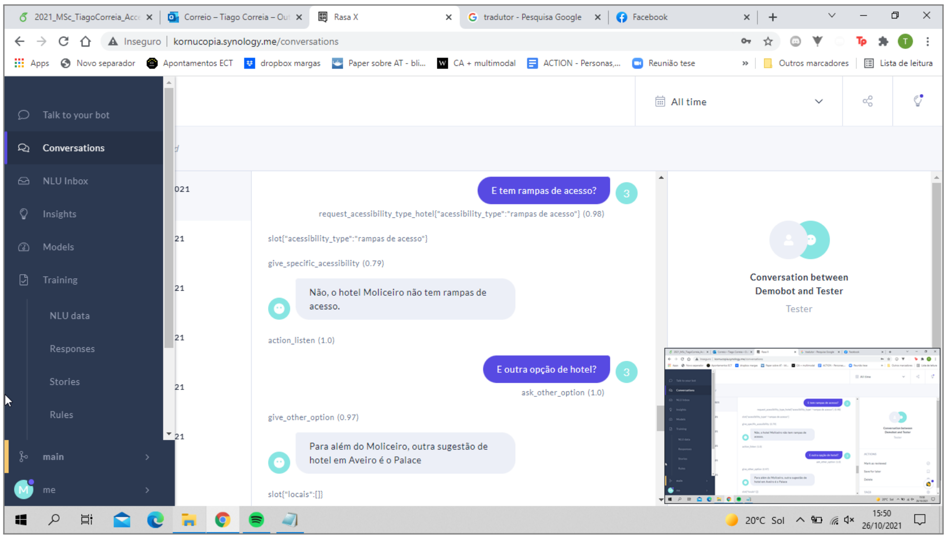
This change also meant that, in the stories file it was necessary to add the entity LOC recognized by spaCy in addition to the `slot_was_set`.

5.4 Use of the improved version by domain experts

Again, RASA X was used for the deployment of this version of the assistant. The conversations with the assistant are made available and were, once again, explored and annotated in order to take full advantage of this feature. The interface for analyzing the conversation in RASA X is shown below: the tab **conversations**, where it is possible to see the full conversation: the recognized intents along with the entities as well and the percentage of confidence of the intent. Another important thing is the **NLU Inbox** tab where it is possible to add the text examples in the user conversations to an intent or correct the recognized intent of an user utterance and add it as an example as well.

The next step was to make available to a larger number of researchers in ACTION team this improved version of the assistant. Unlike in the first evaluation of the conversational assistant, now the purpose was to see if the assistant was already useful and corresponded to the request of the users, but as well it was important to get information about how it could still typical evolve.

To this end, domain experts were informed about the overall context and motivations considered for developing the assistant, the extent of the features supported and were provided with a list of suggestions of actions that they could perform as a starting point for interacting with the assistant. The participants were, once again, instructed to use natural speech and, after testing the suggested action, to move into obtaining other information judged pertinent for PwD.



E tem rampas de acesso?

request_acessibility_type_hotel{"acessibility_type":"rampas de acesso"} (0.98)

slot{"acessibility_type":"rampas de acesso"}

give_specific_acessibility (0.79)

Não, o hotel Moliceiro não tem rampas de acesso.

action_listen (1.0)

E outra opção de hotel?

ask_other_option (1.0)

give_other_option (0.97)

Para além do Moliceiro, outra sugestão de hotel em Aveiro é o Palace

slot{"locais":[]}

Figure 5.2: Example of the RASA X interface to analyze conversations.

To take the most out of the participants' inputs, like it occurred in the first evaluation of the assistant, the conversations were continuously analyzed, typically everyday so all the participants could see improvements and didn't encounter the same problem over and over again, to understand the behavior of the assistant and in what ways should the assistant improve in order to fulfill the questions/intents provided by the testers. This resulted in novel data being annotated or corrected, and new intents and actions added resulting in novel models. With this approach it was noticeable, to the participants, that the assistant was actually improving, with their participation.

5.4.1 Testing Outcomes and Discussion

As already expected, users asked for an hotel with access to wheelchairs but also with other types of accessibility like hotels with information in braille, hotels with ramps throughout the course. Also, users tried to ask for museums and restaurants with some type of accessibility as well which lead to create intents and actions to deal with this matter. Another small problem but that had a lot of impact was that whenever an user said "can you find me a place to stay in Aveiro", the assistant couldn't understand that that sentence refers to a request for an hotel and that sentence was added to the NLU under the intent `request_hotel`.

Overall, the added intents were **`request_museum_with_access`** and **`request_restaurant_with_access`** and the corresponding actions **`give_museum_with_access`** and **`give_restaurant_with_access`**. Also, as explained above, some sentences used in conversations were added to the NLU inbox to specific intents like `request_hotel` and `ask_for_other_option` to make it easier to recognize those intents.

5.5 Conclusions

In this chapter is described the first efforts in creating a Conversational Assistant for Accessible Tourism. For this context it was used RASA to develop a first version of the assistant and, after, instantiate it with RASA X, following a CDD methodology. The development of the Conversational Assistant was an iterative process and domain experts were included in the task of refinement and improvement of the assistant. Throughout the course the recognition of entities such as cities was improved using spaCy to do it.

It was also really important that RASA framework included RASA X as a feature, not only because it allowed the Conversational Assistant to follow the iterative process along the way, but also because, by allowing domain experts to interact with the assistant throughout the development, brought a special dynamic to the development of the Conversational Assistant. These interactions contributed, on a large scale, to the improvement and definition of same features of the Conversational Assistant. Without that, the task of including domain experts in the development of the assistant would be much harder and the quality of the assistant would decrease, for sure.

CHAPTER 6

Conclusions

In short, the work performed enlights the first efforts in creating a Conversational Assistant for Accessible Tourism.

Having been adopted an user centered design methodology, a set of four personas were validated, describing the different disabilities and their motivations, using the questionnaires and focus groups carried out under the research project ACTION, After that, scenarios were constructed in order to demonstrate the usage of the system by those personas, already taking into account that the technological solution being used was a conversational agent and adapting the scenarios to the conversation-style commonly used with assistants. From that same scenarios, the requirements were extracted with different levels of priority: **P0** for the most important ones to develop and **P2** for the ones only to develop if there was time.

After that, the first version of the assistant was developed in RASA with a subset of requirements of the ones extracted in the scenarios (functionalities). Following a CDD methodology, which states that is best to develop an assistant with a small set of functionalities and use the evaluation to improve continuously the assistant and the set of functionalities, the assistant was deployed in RASA X. The link with the assistant generated was shared with two researchers in the ACTION project and, everyday, their conversations were analyzed in order to improve the assistant. That resulted in a second version of the assistant with new functionalities: using the evaluation of the first assistant and bringing other requirements that were not taken into account for the first version.

+

Although that, the assistant can still improve in several aspects such as:

- voice: it was a requirement that wasn't fulfilled but it is important that the assistant can understand and say things in both text and voice, specially because, for example, for blind people it is so important, not only to be able to speak and not text but also that the incoming messages from the assistant be through voice and not text.
- integration with other channels

- photos: provide photos of places, so the person can understand and foresee the problems that will face visiting or staying in that place.
- feedback: add feedback of other people with the same disability or other regarding an hotel, museum or other local.

´

APPENDIX A

Materials Used To Contextualize the Evaluation

Assistente para Turismo Acessível

1º versão

Tiago Jorge Costa Correia nº80378

Orientadores: professor Samuel Silva e professor António Teixeira

Universidade de Aveiro

Turismo Acessível

- O acesso à informação sobre a acessibilidade de diferentes locais e pontos de interesse é um aspeto importante no domínio do Turismo Acessível.

- A quantidade de informação que se pretende disponibilizar assim com as características muito diversas dos potenciais interessados criam desafios de como a disponibilizar.

- Um **assistente conversacional** pode disponibilizar um modo alternativo às interfaces gráficas tradicionais, podendo o utilizador interagir conversando com um assistente.

Assistente Conversacional para Turismo Acessível

O **objetivo** desta primeira fase de utilização é permitir que o assistente possa conversar com mais pessoas de modo a poder ter acesso a diferentes formas de conversação e a questões que podem ser relevantes para ele saber responder, no futuro.

- Deverá receber um link que lhe vai permitir conversar com o assistente.
- Ser-lhe-á sugerida, mais à frente nestes slides, alguma informação a ser obtida conversando com o assistente para ajudar a perceber o que podem ser as conversas
- Para além destas tarefas pode, posteriormente, falar livremente com o assistente.

Capacidades do assistente (para já)

- Neste momento, o assistente consegue responder a perguntas sobre hotéis, museus e locais de interesse para uma determinada cidade, perguntas genéricas sobre a acessibilidade destes locais, se um determinado hotel é acessível para cadeira de rodas e pedir outras sugestões quer de hóteis, museus ou locais de interesse.
- Note que os dados disponíveis sobre estes elementos são simulados e destinam-se a ilustrar as possibilidades do assistente.

Acesso online ao assistente

- Pode aceder ao assistente seguindo este link:

http://kornucopia.synology.me/guest/conversations/production/f717872aad9d453d8cd3a6a6c5639a5f

§ Quando o assistente não responder ou quando acha que o assistente não respondeu convenientemente à sua pergunta, pode responder ao assistente diretamente, dizendo como deveria ter respondido, por exemplo **"Assistente, devias ter respondido..."**

Sugestão de informação a obter do assistente

Ao conversar com o assistente pode tentar obter informação de acordo com as sugestões abaixo. Converse com o assistente com a maior naturalidade possível.

- Pedir um hotel para um determinado local e obter a informação sobre a sua acessibilidade. Depois, encontrar um museu para visitar.
- Pedir um museu num certo local, seguido da sua acessibilidade e, no final, encontrar uma alternativa.
- Pedir um hotel num certo local, pedir outra sugestão e saber se esse hotel tem acesso para cadeira de rodas.
- **OS LOCAIS TÊM DE COMEÇAR COM LETRA MAIÚSCULA (Braga, Porto, Lisboa, Leiria...)**

Utilização livre do assistente

- No final, experimentar tudo o que vos pareça fazer sentido para o domínio de Turismo Acessível, até novas capacidades que achem que o assistente deva ter em termos de acessibilidade. Se o assistente não responder convenientemente à sua pergunta, sugira, no formulário, uma resposta possível a essa pergunta.

- Quando o assistente não responder ou quando acha que o assistente não respondeu convenientemente à sua pergunta, pode responder ao assistente diretamente, dizendo como deveria ter respondido, por exemplo **"Assistente, devias ter respondido...",** ou então deixar a sugestão e a respetiva resposta no formulário.

Formulário

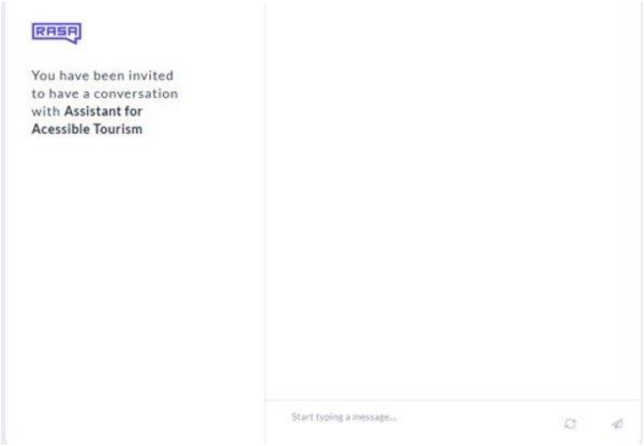
- Depois de ter utilizado o assistente e se não tiver mais sugestões, aceda ao seguinte formulário para responder à sua experiência com o assistente.

https://forms.gle/74KnCreFXLhQFJpz7

Dúvidas e/ou sugestões

- Qualquer dúvida pode contactar-me por e-mail para tcorreia@ua.pt.
- Não se esqueça de preencher o forms com todas as sugestões que tiver!
- Obrigado, desde já, pela participação :)

Interface do RASA-X



You have been invited to have a conversation with **Assistant for Acessible Tourism**

- O primeiro botão do lado esquerdo serve para fazer restart da conversa.
- Podem usá-lo sempre que pretender iniciar uma nova conversa ou se achar que o assistente "se perdeu" na conversa.

References

[1] S. Darcy and T. J. Dickson, “A whole-of-life approach to tourism: The case for accessible tourism experiences,” *Journal of Hospitality and Tourism Management*, vol. 16, no. 1, pp. 32–44, 2009.

[2] S. McCabe and A. Diekmann, “The rights to tourism: Reflections on social tourism and human rights,” *Tourism Recreation Research*, vol. 40, no. 2, pp. 194–204, 2015.

[3] A. N. Evcil, “Barriers and preferences to leisure activities for wheelchair users in historic places,” *Tourism Geographies*, vol. 20, no. 4, pp. 698–715, 2018.

[4] W. H. Organization *et al.*, *International classification of impairments, disabilities, and handicaps: a manual of classification relating to the consequences of disease, published in accordance with resolution WHA29. 35 of the Twenty-ninth World Health Assembly, May 1976.* World Health Organization, 1980.

[5] I. Ine, “Censos 2011 resultados definitivos-portugal,” *Lisboa-Portugal: Instituto Nacional de Estatıstica, IP*, 2012.

[6] D. Buhalis and S. Darcy, *Accessible tourism: Concepts and issues.* Channel View Publications, 2010.

[7] J. A. Card, S. T. Cole, and A. H. Humphrey, “A comparison of the accessibility and attitudinal barriers model: Travel providers and travelers with physical disabilities,” *Asia Pacific Journal of Tourism Research*, vol. 11, no. 2, pp. 161–175, 2006.

[8] N. M. Ray and M. E. Ryder, “
ebilities” tourism: An exploratory discussion of the travel needs and motivations of the mobility-disabled,” *Tourism Management*, vol. 24, no. 1, pp. 57–72, 2003.

[9] I. Bauer, “When travel is a challenge: Travel medicine and the ‘dis-abled’traveller,” *Travel medicine and infectious disease*, vol. 22, pp. 66–72, 2018.

[10] T. Mccoy, “Law enforcement understanding their impact and building a better rapport with people with disabilities (pwd),” 2020.

[11] A. Zajadacz, “Sources of tourist information used by deaf people. case study: The polish deaf community,” *Current Issues in Tourism*, vol. 17, no. 5, pp. 434–454, 2014.

[12] E. Devile and E. Kastenholz, “Accessible tourism experiences: The voice of people with visual disabilities,” *Journal of Policy Research in Tourism, Leisure and Events*, vol. 10, no. 3, pp. 265–285, 2018.

[13] B. Gillovic, A. McIntosh, S. Darcy, and C. Cockburn-Wootten, “Enabling the language of accessible tourism,” *Journal of Sustainable Tourism*, vol. 26, no. 4, pp. 615–630, 2018.

[14] S. Beart, D. Hawkins, B. S. Kroese, P. Smithson, and I. Tolosa, “Barriers to accessing leisure opportunities for people with learning disabilities,” *British Journal of Learning Disabilities*, vol. 29, no. 4, pp. 133–138, 2001.

[15] F. A. McGuire, “A factor analytic study of leisure constraints in advanced adulthood,” 1984.

[16] M. J. Daniels, E. B. D. Rodgers, and B. P. Wiggins, “
travel tales”: An interpretive analysis of constraints and negotiations to pleasure travel as experienced by persons with physical disabilities,” *Tourism management*, vol. 26, no. 6, pp. 919–930, 2005.

[17] M. Horgan-Jones and L. Ringaert, “Accessible tourism in manitoba,” *Niagara Falls, Canada: Travel and tourism research Association*, 2001.

[18] N. J. Stumbo and S. Pegg, “Travelers and tourists with disabilities: A matter of priorities and loyalties,” *Tourism Review International*, vol. 8, no. 3, pp. 195–209, 2005.

[19] B. McKercher, T. Packer, M. K. Yau, and P. Lam, “Travel agents as facilitators or inhibitors of travel: Perceptions of people with disabilities,” *Tourism Management*, vol. 24, no. 4, pp. 465–474, 2003.

[20] C. Brown, “Assistive technology computers and persons with disabilities,” *Communications of the ACM*, vol. 35, no. 5, pp. 36–45, 1992.

[21] K. L. Southwell and J. Slater, “An evaluation of finding aid accessibility for screen readers,” *Information Technology and Libraries*, vol. 32, no. 3, pp. 34–46, 2013.

[22] C. Martins, “Museum audio guides as an accessibility enhancer,” *Accesibilidad en la nueva era de las comunicaciones Profesionales y universidad: un diálogo imprescindible*, pp. 101–115, 2012.

[23] H. Rumetshofer and W. Wöß, “Tourism information systems promoting barrier-free tourism for people with disabilities,” in *International Conference on Computers for Handicapped Persons*, Springer, 2004, pp. 280–286.

[24] J. Lester, K. Branting, and B. Mott, “Conversational agents,” *The Practical Handbook of Internet Computing*, pp. 220–240, 2004.

[25] D. R. Traum and E. A. Hinkelman, “Conversation acts in task-oriented spoken dialogue,” *Computational intelligence*, vol. 8, no. 3, pp. 575–599, 1992.

[26] N. M. Radziwill and M. C. Benton, “Evaluating quality of chatbots and intelligent conversational agents,” *arXiv preprint arXiv:1704.04579*, 2017.

[27] J.-G. Harms, P. Kucherbaev, A. Bozzon, and G.-J. Houben, “Approaches for dialog management in conversational agents,” *IEEE Internet Computing*, vol. 23, no. 2, pp. 13–22, 2018.

[28] V. K. Cannanure, T. X. Brown, and A. Ogan, “Dia: A human ai hybrid conversational assistant for developing contexts,” in *Proceedings of the 2020 International Conference on Information and Communication Technologies and Development*, 2020, pp. 1–5.

[29] T.-H. Huang, J. C. Chang, and J. P. Bigham, “Evorus: A crowd-powered conversational assistant built to automate itself over time,” in *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, 2018, pp. 1–13.

[30] N. Ventoura, K. Palios, Y. Vasilakis, G. Paraskevopoulos, N. Katsamanis, and V. Katsouros, “Theano: A greek-speaking conversational agent for covid-19,” in *Proceedings of the 1st Workshop on NLP for Positive Impact*, 2021, pp. 36–46.

[31] D. Griol, Z. Callejas, and D. Pérez-Fernández, “Hispabot-covid19: The official spanish conversational system about covid-19,” *Proc. IberSPEECH 2021*, pp. 139–142, 2021.

[32] S. Siedlikowski, L.-P. Noël, S. A. Moynihan, M. Robin, *et al.*, “Chloe for covid-19: Evolution of an intelligent conversational agent to address infodemic management needs during the covid-19 pandemic,” *Journal of Medical Internet Research*, vol. 23, no. 9, e27283, 2021.

[33] S. Ultes, L. M. Rojas-Barahona, P.-H. Su, D. Vandyke, D. Kim, I. Casanueva, P. Budzianowski, N. Mrkšić, T.-H. Wen, M. Gašić, and S. Young, “PyDial: A multi-domain statistical dialogue system toolkit,” in *Proceedings of ACL 2017, System Demonstrations*, Vancouver, Canada: Association for Computational Linguistics, Jul. 2017, pp. 73–78. [Online]. Available: `https://www.aclweb.org/anthology/P17-4013`.

[34] J. D. Finch and J. D. Choi, *Emora STDM: A versatile framework for innovative dialogue system development*, 2020. arXiv: `2006.06143 [cs.CL]`.

[35] A. Papangelis, M. Namazifar, C. Khatri, Y.-C. Wang, P. Molino, and G. Tur, *Plato dialogue system: A flexible conversational ai research platform*, 2020. arXiv: `2001.06463 [cs.HC]`.

[36] P. Lison and C. Kennington, “Opendial: A toolkit for developing spoken dialogue systems with probabilistic rules,” in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Demonstrations)*, Jan. 2016, pp. 67–72. DOI: `10.18653/v1/P16-4012`.

[37] V. L. Shiv, C. Quirk, A. Suri, X. Gao, K. Shahid, N. Govindarajan, Y. Zhang, J. Gao, M. Galley, C. Brockett, T. Menon, and B. Dolan, "Microsoft ICECAPS: An open-source toolkit for conversation modeling," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 123–128. DOI: `10.18653/v1/P19-3021`. [Online]. Available: `https://www.aclweb.org/anthology/P19-3021`.

[38] T. Bocklisch, J. Faulkner, N. Pawlowski, and A. Nichol, "Rasa: Open source language understanding and dialogue management," *arXiv preprint arXiv:1712.05181*, 2017.

[39] R. K. Sharma and M. Joshi, "An analytical study and review of open source chatbot framework, rasa," *International Journal of Engineering Research and*, vol. 9, no. 06, 2020.

[40] C. Abras, D. Maloney-Krichmar, J. Preece, *et al.*, "User-centered design," *Bainbridge, W. Encyclopedia of Human-Computer Interaction. Thousand Oaks: Sage Publications*, vol. 37, no. 4, pp. 445–456, 2004.

[41] T. Adlin, J. Pruitt, K. Goodwin, C. Hynes, K. McGrane, A. Rosenstein, and M. J. Muller, "Putting personas to work," in *CHI'06 Extended Abstracts on Human Factors in Computing Systems*, 2006, pp. 13–16.

[42] T. Miaskiewicz and K. A. Kozar, "Personas and user-centered design: How can personas benefit product design processes?" *Design studies*, vol. 32, no. 5, pp. 417–430, 2011.

[43] J. Pruitt and J. Grudin, "Personas: Practice and theory," in *Proceedings of the 2003 conference on Designing for user experiences*, 2003, pp. 1–15.

[44] P. Rebaque-Rivas, E. P. Gil-Rodrıguez, and I. Manresa-Mallol, "Mobile learning scenarios from a ucd perspective," in *Proceedings of the 12th international conference on Human computer interaction with mobile devices and services*, 2010, pp. 389–390.

[45] B. A. Nardi, "The use of scenarios in design," *ACM SIGCHI Bulletin*, vol. 24, no. 4, pp. 13–14, 1992.

[46] S. R. Haynes, S. Purao, and A. L. Skattebo, "Situating evaluation in scenarios of use," in *Proceedings of the 2004 ACM conference on Computer supported cooperative work*, 2004, pp. 92–101.

[47] K. Wiegers and J. Beatty, *Software requirements.* Pearson Education, 2013.

[48] N. Seyff, N. Maiden, K. Karlsen, J. Lockerbie, P. Grünbacher, F. Graf, and C. Ncube, "Exploring how to use scenarios to discover requirements," *Requirements Engineering*, vol. 14, no. 2, pp. 91–111, 2009.