

Article

# DisBot: A Portuguese Disaster Support Dynamic Knowledge Chatbot

João Boné <sup>1,2,\*</sup> , João C. Ferreira <sup>1,2,3</sup> , Ricardo Ribeiro <sup>1,4</sup>  and Gonçalo Cadete <sup>2,5</sup> 

<sup>1</sup> Instituto Universitário de Lisboa (ISCTE-IUL), 1649-026 Lisbon, Portugal; jcafa@iscte-iul.pt (J.C.F.); ricardo.ribeiro@iscte-iul.pt (R.R.)

<sup>2</sup> INOV INESC Inovação—Instituto de Novas Tecnologias, 1000-029 Lisbon, Portugal; goncalo.cadete@tecnico.ulisboa.pt

<sup>3</sup> Instituto Universitário de Lisboa (ISCTE-IUL), ISTAR, 1649-026 Lisbon, Portugal

<sup>4</sup> INESC-ID Lisboa, 1000-029 Lisbon, Portugal

<sup>5</sup> Instituto Superior Técnico, Universidade de Lisboa, 1049-001 Lisbon, Portugal

\* Correspondence: jmbbe@iscte-iul.pt

Received: 28 November 2020; Accepted: 11 December 2020; Published: 18 December 2020



**Abstract:** This paper presents DisBot, the first Portuguese speaking chatbot that uses social media retrieved knowledge to support citizens and first-responders in disaster scenarios, in order to improve community resilience and decision-making. It was developed and tested using Design Science Research Methodology (DSRM), being progressively matured with field specialists through several design and development iterations. DisBot uses a state-of-the-art Dual Intent Entity Transformer (DIET) architecture to classify user intents, and makes use of several dialogue policies for managing user conversations, as well as storing relevant information to be used in further dialogue turns. To generate responses, it uses real-world safety knowledge, and infers a dynamic knowledge graph that is dynamically updated in real-time by a disaster-related knowledge extraction tool, presented in previous works. Through its development iterations, DisBot has been validated by field specialists, who have considered it to be a valuable asset in disaster management.

**Keywords:** disaster management; community resilience; situational awareness; natural language processing; chatbots; knowledge graphs

## 1. Introduction

Disaster management is a series of procedures intended to be implemented before, during and after disasters, in order to avoid or minimize their damage [1]. It is becoming a wide and more spread out topic, as we have seen an increase in disaster occurrences, whether from extreme weather events, such as earthquakes, hurricanes or floods, or from man-made disasters like most wildfires and terrorism. According to Bernstein [2], the demand for risk management has risen along with the growing number of risks we are facing, which directly leads to an urging need for advancements in disaster management technologies. However, the use of automatic information dissemination technologies, such as chatbots, is still very lacking in this field of study, although their potential for improving authorities' decision making and population access to information, is substantial [3].

Language is the cornerstone of human communication and sentience, and conversation is the most basic and uniquely privileged domain of that cornerstone. It is the first kind of language we know as children, and for the majority of us, it is the kind of language we most commonly nourish [4]. Chatbots are machine agents that serve as natural language user interfaces for data and service providers [5]. They make use of our attachment to natural language conversations to enable users to have a greater sense of security when exchanging information with an artificial intelligence (AI) entity.

They achieve this by closely simulating what a conversation with another human being would be like, in a specific scenario.

Chatbots have been around for a while as both personal assistants and information facilitators. We saw their birth in 1966, when Weizenbaum [6] created a “system which makes certain kinds of natural language conversation between man and a computer possible”. His creation, Eliza, analyzed input sentences on the basis of keywords. Chatbots have since come a long way, from Parry in the 1970s to the focus in task-completion virtual assistants in the 2000s. From 2015 on, chatbot research showed an exponential growth, displaying a progress from virtually no publications in the year 2000, to thousands of publications in 2019. Personal assistants and social chatbots have been, by far, the main focus of this field’s applications in the 2000s and 2010s, led by huge developments in personal assistants of big companies, such as Alexa and Siri, and the recent social chatbots, such as Microsoft’s XiaoIce [7]. These entities have, nevertheless, the tremendously challenging task of efficiently working in open domain scenarios [7]. In closed domain scenarios, however, chatbots have also been widely used. In the healthcare sector, they have been used in aiding treatment for different diseases such as cancer [8] and asthma [9], or even for promoting healthier habits [10]. Chatbots are also being used in several other closed domain trending areas, such as home automation [11], customer service [12,13], and education [14]. As seen in these sectors of activity, chatbots have a huge potential for disseminating information that is, in many different ways, beneficial to their users. With all of this in mind, in disaster-management, which is also a closed-domain, chatbots have the potential to efficiently disseminate information that will positively impact both authorities and population’s decision making.

The state-of-the-art in disaster support chatbot systems is very recent and solely focused on decision makers, and formal knowledge such as official reports, for a very limited amount of disasters. There is an opportunity of creating a system that can take advantage of the large amounts of data human beings are constantly creating. The relevance of this system can be further enhanced by being able to summarize this data and inform disaster stakeholders about it, in an automatic and reliable way, which would greatly improve human resilience and facilitate decision making in disaster-related emergency scenarios. On top of this, using technologies and knowledge structures that are easily built upon would further enhance the value of this system and contribute to disaster management.

Our main objective, with this work, is to devise an artifact that consists of a Portuguese disaster support chatbot system with the capacity to use DisKnow [15]—a previous work on disaster support knowledge extraction and representation—and to inform its users with knowledge about detected disasters and how to better deal with them. DisBot builds upon the state-of-the-art not only by the recent technologies it uses, but also by managing several different types of disasters, as opposed to specific water-related disasters only [16–18]. It is also the first disaster support chatbot using the Portuguese language, which—despite being the sixth most spoken language in the world, with over 260 million speakers—still has very few available resources when compared to other widely spoken languages, such as English [19].

With this in mind, our work presents a small-scale systematic review on disaster support chatbots. Following the Design Science Research Methodology (DSRM) process, DisBot’s pipeline is then presented, and is detailed for the first iteration of its development. Afterward, an overview of the validation of the first iteration is provided, as well as the development and validation of the two smaller remaining iterations. Finally, the conclusions and future work of our development are discussed.

## 2. Related Work

Chatbots have enormous potential as knowledge disseminators in time-critical scenarios, such as disasters, where every action must be well informed and taken as fast as possible. Despite their latent potential, Chatbots applied to disaster support are still very much unknown to the scientific community.

We have led a systematic review using Scopus (<https://www.scopus.com/home.uri>) as our primary research database, and Google Scholar (<https://scholar.google.com/>) as a secondary source for papers, which required a more restrictive approach, due to the range of quality of the works it

contains. Initially, we have used a very strict search query, which led to very few results, out of which close to none were relevant. This initial query was the following:

*(chatbot OR “virtual assistant”) AND (crisis OR hazard OR catastrophe OR disaster)*

Seeing the lack of achieved results, and taking into account the need for a deeper research on this topic, we have redrafted our research query with more synonyms of the words we had used before, and adding relevant types of disasters. Hence, the review was conducted using the following search query:

*(chatbot OR chatterbox OR “chat agent” OR “conversational system” OR “conversational agent” OR “conversational interface” OR “question answering” OR “virtual assistant” OR “virtual agent”) AND (crisis OR hazard OR “humanitarian aid” OR emergency OR catastrophe OR disaster OR havoc OR calamity OR “extreme weather events” OR fire OR flood OR earthquake)*

The search string inquiry returned 163 studies, which were then filtered according to three phases: filtering criteria, abstract screening, and full-text screening. For the first of these filtering phases we set an assortment of criteria related to the content of the work, their type of publication, and their availability. Table 1 displays all of these criteria.

**Table 1.** Filtering criteria for the systematic review.

Filtering Criteria
Not written in English nor Portuguese
Publication date before 2010
Paid documents outside of our licenses or documents published uniquely in non-reliable sources
No title relevance in respect of chatbots and disaster support

Table 2 shows the results of all filtering stages, numbering the selected studies for each of them.

**Table 2.** Study selection for the systematic review.

Filtering Phase	Number of Studies
Search string inquiry	163
Inclusion and exclusion criteria	23
Abstract screening	9
Full-text screening	3

Through this research, we have come across the few existing relevant instances of chatbots applied to disaster situations with a supportive role.

In 2018, Sermet and Demir [16] created a flood support chatbot system named Flood AI. This system uses a microservices-oriented architecture, in which each module acts as an autonomous service, but the system, as a whole, aims to offer stakeholders information regarding flood preparedness and response. Concerning its knowledge, it covers a month’s worth of data—14 days past and 10 days future—accessed from the Iowa Flood Information System, which includes flood inundation maps, real-time flood conditions, forecasts, and others. When the system receives a user’s question, it uses an ontology to extract useful information, such as the location, date and time, and intent. The Natural Language Understanding (NLU) module also uses a third-party spell checker, which presents potential spelling mistakes and suggestions on how to correct them. After its understanding, the question is mapped to one of the question models of the system. These models were created with pre-assigned weights and aim to provide the system with information regarding which databases to use, which analyses to apply, and the format in which the answer should be provided. Such models assure the system greater flexibility when adding support to new questions, without the need for

in-depth computer science experience. Lastly, the answer can be given to the user via natural language, or through images such as graphs and maps. Even though the system is already quite complex and complete, the authors refer that the user interaction can be further enhanced, by asking clarification questions, when the system is unable to extract enough information from a user's question.

One year later, in 2019, Tsai et al. [17] have proposed a water-related disaster support chatbot system named Ask Diana, which has ties with Taiwan's Water Resource Agency. The purpose of this system is to help decision making in flood or drought scenarios, by effectively presenting decision makers with official data reports, such as weather reports, meteorograms, disaster response reports and others. As for its knowledge, the system uses a database consisting of pre-gathered and cataloged reports. To access this data, a user can both manually navigate through a menu panel, or input the system with text. This text is composed of a set of keywords that the system then matches with a static built-in mapping table, as to understand the user's intent. Afterward, the system uses a fuzzy search method, consisting of a decision tree. The most related report is then presented to the user via images or, if there are several reports sharing the same score, the top four are shown. Although the system has displayed promising results in its usability test, the authors have reported some system limitations, such as its lack of NLP algorithms for both understanding user requests—which is done by comparing input keywords with a built-in mapping table—and presenting information. Another constraint the users have mentioned is the labor-intensive maintenance of the system, as the keyword mapping table is handcrafted and already contains up to 200 keywords, making further updates very time-consuming.

Also in 2019, one of the authors of [17] led the development of a new system [18]. Although implemented in the same domain, this system aims at correcting some of Ask Diana's faults. It uses a MongoDB database with both static and dynamic information, the first being collected from documents and tables provided from local governments, and the last being weather data that is streamed online. In its process, the system first performs an analysis on the user's input text. It starts by detecting the class of the question and then proceeds to use an ontology to parse and detect both the target of the question and information aiding its querying. This information is then sent to a search module, which is subdivided into two separate functions: path-planning and query formation. Path planning adopts Dijkstra's shortest path algorithm to find the shortest path to the information the users want and, even though the authors do not provide much information about the formation of the path, they do refer that all distances between edges were set to one. The second function of the search module, query formation, was designed to transform the planned path into the exact query to be executed in the knowledge base. Regarding the answer provided to the user, it is based on natural language, and uses sentence patterns to fit the gathered data into natural language sentences. The authors also refer that there are some aspects in which the system needs improvement, the main one being its lack of consideration for more complicated search tasks, and is directly related to the path-planning phase of the process.

From the more than a hundred studies that resulted from our research, only three showed enough significance to be considered. We believe it is very significant, as all of these studies were published in either 2018 or 2019. This shows us that even though chatbots and disaster-management are thriving research areas, their blend is still quite uncharted and open to new contributions, showing a large range of opportunities. Table 3 presents a summary and comparison of these chatbot systems.

**Table 3.** Summary of the systems included in the systematic review.

Paper	Year	Lang	Disasters	Knowledge	Natural Language Understanding	Knowledge Lookup	Replies
Sermet and Demir [16]	2018	EN	Floods	API to a web-based platform	Ontology-based extraction	Question models with pre-assigned weights	Images, Natural Language
Tsai et al. [17]	2019	ZH	Floods Droughts	Pre-classified static files database MongoDB with static and dynamic information	Keyword-based mapping table	Fuzzy search decision tree	Images
Chan and Tsai [18]	2019	ZH	Floods Droughts	GraphDB with static and dynamic information	Ontology-based extraction	Path planning and query formation	Natural Language
DisBot	2020	PT	Floods Fires Explosions Earthquakes	GraphDB with static and dynamic information	Ontology-based extraction	Path planning and query formation	Natural Language

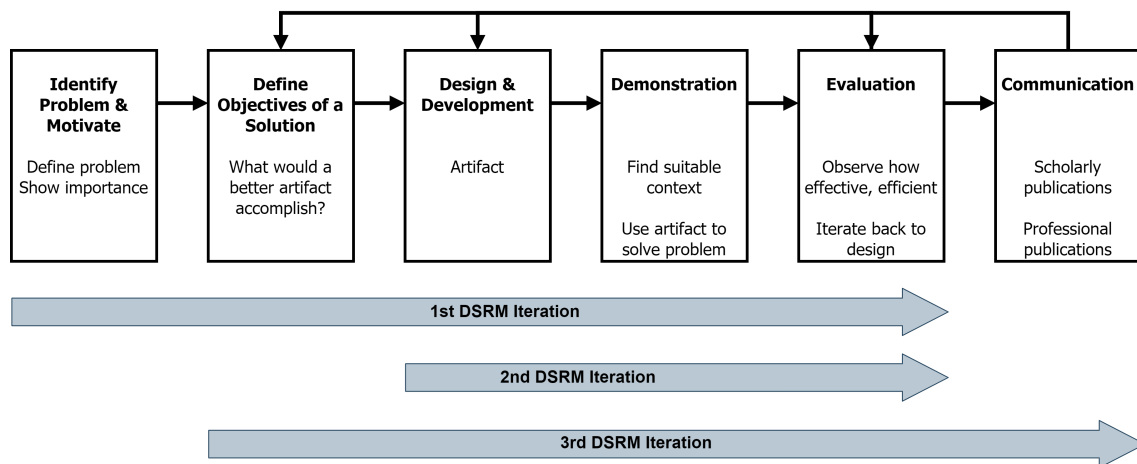
The three mentioned studies aimed at providing disaster-management stakeholders with very specialized information regarding water-based disasters. Even though their means of achieving such results differed greatly, the use of ontology-based extraction of information from user questions seems to help achieve better results. The keyword-based mapping table implemented in [17] has limited the possible interactions of their system due to the potential and flexibility of chatbot interaction through natural language. The integration of both static and dynamic information in a knowledge base has also positively impacted results in [16,18], as disaster management requires both real-time updated data about disasters taking place, and predefined information on how to deal with specific scenarios.

### 3. Methodology

Without a strong component that provides relevant research solutions, Information Systems (IS) research faces a possible lack of control on the fields for which its applicability is of significant importance [20]. There are two key paradigms that characterize IS research. On the one hand, we have behavioral science, which attempts to establish hypotheses that predict individual or organizational actions. On the other hand, there is design-science, which aims to expand human and organizational capacities through the development of innovative artifacts [21]. Taking this into account, this work follows the Design Science Research Methodology (DSRM), along with the seven guidelines proposed by Hevner et al. [21]. This methodology has its origins in engineering and artificial sciences, and its main objective is creating relevant artifacts with the clear purpose of adding value to the fields they are applied to.

Since DSRM follows a problem-solving approach, it is important to make assessments of the artifacts to provide input and a clearer understanding of their issues, with a focus on enhancing both their quality and design in the next iterations of the process. This build-and-assess loop is usually iterated a number of times before the final artifact is created [22]. Our work consists of three iterations of the DSRM process model. The first iteration incorporates the majority of this work, going from the initial entry point, to the design and development of the first version of the artifacts, as well as their evaluation. As this methodology promotes interaction with its end-users as the development goes on, the entry points of the second and third iterations of the process were only decided during the conclusion and evaluation of their previous iteration.

Figure 1 pertains to the iterations of this work represented in a nominal sequence consisting of six activities that, according to its authors, resumes the DSRM process.



**Figure 1.** Iterations represented in the Design Science Research Methodology (DSRM) process model, Peffers et al. [20].

According to the methodology, one initial meeting was held to define all of the evaluation criteria, as claimed by the objectives of our work. To better understand what possible end-users would expect from a system of this nature, this meeting had the participation of three civil protection specialists from the Portuguese municipality of Barreiro, who were also the evaluators and stakeholder representatives for all three iterations of the DSRM process.

To objectively define the DSRM evaluation criteria we first need to determine the capabilities of our artifact for disaster management and community resilience. With this in mind, we have resorted to Cadete and da Silva’s proposed resilience assessment framework (RAF), which was proposed with the integration of disaster risk management aspects in mind [23]. The framework allows for a disaster risk management approach to crisis management and is aligned with best practices from the National Institute of Standards and Technology (NIST) [24] and Information Systems Audit and Control Association (ISACA) [25] frameworks. Crossing the objective of this work with the referred RAF, we have selected Situational Awareness (SA) and Intelligence Fusion (IF) as the capability to be evaluated in our work.

Taking into account the distinct contributions expected of the proposed artifacts for situational awareness and intelligence fusion, different criteria were defined to evaluate each of them. The first artifact, the knowledge extraction system, does not directly interact with users, which leads to its evaluation criteria being more objective, and related to its performance. As for the opposing second artifact, the digital assistant, its objective lies in user interaction, prompting its evaluation criteria to be more subjective and related to the dialogue fluidity and how well it presents information to its users.

Although DSRM is a homogeneous idea, artifact evaluation is still discussed amongst the community of this methodology, as in the DSRM literature, evaluation criteria are presented in a fragmented or incomplete manner [26]. To overcome this obstacle, we have decided to follow the hierarchical evaluation criteria for IS artifacts proposed by Prat et al. [26]. The objective statements, shown in Table 4, were then created according to both the selected capability and criteria, which serve as the objectives evaluated in the three iterations of the DSRM process. To evaluate these criteria, ratings are given by each evaluator based on evidence showing that the added value of the objective statement has been fulfilled. With this in mind, we have decided to use ISO 15504’s four-point NLPF scale [27], which consists of the following four levels:

- Not Achieved (NA)—(0–15%)
- Partially Achieved (PA)—(15–50%)
- Largely Achieved (LA)—(50–85%)
- Totally Achieved (TA)—(85–100%)



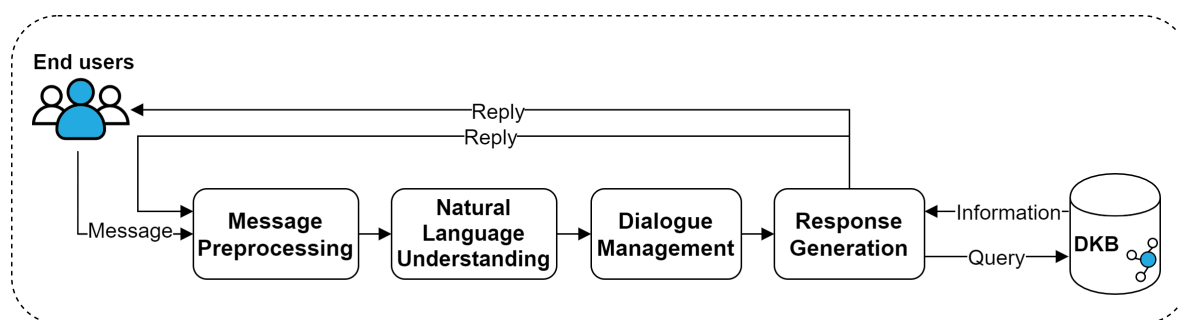
**Table 4.** Objective statements to be used in the DSRM evaluation.

Capability	Dimension	Criteria	Objective Statement
SA & IF	Goal	Efficacy	Effectively leveraging local disaster intelligence to improve citizen and first-responder Situational Awareness and Intelligence Fusion
SA & IF	Environment	Consistency with people/Utility	Enabling information on existing disasters, as well as advice on emergency contacts and procedures
SA & IF	Environment	Consistency with people/Understandability	Providing fluid and understandable human-like conversations (this criterion was only added on the 3rd iteration of the DSRM process)
SA & IF	Environment	Consistency with organization/Utility	Providing disaster management authorities with an alternative and automatic way of informing citizens
SA & IF	Structure	Level of detail	Providing citizens with integrated local knowledge
SA & IF	Evolution	Learning capability	Automatically learning about disaster intelligence and updating that knowledge mid disaster

#### 4. DisBot Framework

The conceived artifact, DisBot, receives messages and sends replies accordingly. In between these happenings, it needs to preprocess the messages it receives, which is done by tokenizing and extracting relevant features from these messages. Both of these actions are done using SpaCy (<https://spacy.io/>).

After the messages have been preprocessed, the chatbot needs to retrieve relevant information from them. This step is called Natural Language Understanding (NLU), and it implies both the extraction of named entities and the classification of the user’s intent, that is, the intention behind the user’s message. Afterward, the Dialogue Management component consolidates the user’s message with the previous steps of the conversation they have had, if any. This consists in understanding if this message is a dialogue turn of a previous story and mapping this message to its appropriate action. The last step on the Digital Assistant’s pipeline consists in generating a response, which can go from simple tasks such as asking for clarification or replying with a predefined message, to custom and more complex actions, such as generating a new response based on the knowledge that is in the Dynamic Knowledge Base (DKB). The chatbot pipeline is illustrated in Figure 2.



**Figure 2.** DisBot framework.

One of the main tools used in building our disaster support chatbot was RASA (<https://rasa.com/>). RASA is a machine learning framework for the automation of text- and voice-based assistants that allows for a higher level of abstraction when creating chatbots. We have specifically used RASA Open Source, the free version of the solution, which, although limited, has enough tools for the development of an effective chatbot. The reasons

we have decided to use RASA over similar and more well-known tools, such as Google’s Dialogflow (<https://cloud.google.com/dialogflow>) or Facebook’s Wit.ai (<https://wit.ai/>), were its very transparent development and its large open source community, which is responsible for giving this framework a very sophisticated NLU engine, as well as the capability of easily integrating external tools. Along with RASA, several other tools have been used, such as SpaCy’s Portuguese language models and preprocessing algorithms, as well as other preprocessing and NLU tools such as scikit-learn (<https://scikit-learn.org/stable/>) featurizers and Facebook’s Duckling (<https://github.com/facebook/duckling>), which is a Haskell library that parses text into structured data.

#### 4.1. Message Preprocessing

Message preprocessing is required for preparing the textual data the chatbot receives—both user questions and previous chatbot replies—to be fed to its models. The first is tokenizing the data, which is done using SpaCy’s tokenizer. This tokenizer follows language-specific sets of rules to segment the text into words, punctuation and so on. As the full text and its structure are important for certain NLP tasks that are relevant for chatbots, such as named-entity recognition, we have not used other common preprocessing methods, such as stop-word removal, lemmatization, etc. The second and last step is the featurization. This step consists in transforming the tokenized text into numerical features that the NLU models can understand and consume. For this task we have used several featurizers with different purposes. Table 5 specifies all of these featurizers and their usage.

**Table 5.** Used featurizers and related functions.

Featurizer	Type	Function
Spacy Featurizer	Dense	Entity extraction
		Intent classification
		Response classification
Lexical Syntactic Featurizer	Sparse	Entity extraction
scikit-learn Count Vectorizer (words)	Sparse	Intent classification
		Response selection
scikit-learn Count Vectorizer (character n-grams)	Sparse	Intent classification
		Response selection

#### 4.2. Natural Language Understanding

The NLU component of DisBot aims to understand/identify two types of essential information: intents and entities. The first can be interpreted as understanding what the user wants, that is, what is the intention of his/her message, and can be as simple as classifying a “Olá amigo!/Hey Friend!” as a greeting. The second one, however, aims to help the chatbot detail the user’s intention, using custom entities, as some intents are dependent on additional information to be handled. A simple example would be “Como posso falar com os bombeiros?/How can I speak with the fire brigade?”, which, for example, purposes can be interpreted as an `ask_for_contact` intent. Without identifying “bombeiros/fire brigade” as a type of organization the chatbot would never have been able to answer that question correctly, as it would have no means to understand whose contact the user needed.

Our NLU component has been trained using more than 400 Portuguese hand-made annotated phrases that we have created, divided into 25 distinct intents. These phrases also include more than 100 distinct entity references for four distinct custom entities: time, location, organization, and disaster type. Our intents are divided into generic intents, chitchat intents, and request intents. Table 6 presents all of these intents, with some examples.



**Table 6.** Chatbot implemented intents with unannotated examples.

Category	Intent	Unannotated Example	English Equivalent
Generic	greet	Olá	Hello
	goodbye	até breve	see you later
	thanks	obrigado pela ajuda	thank you for the help
	affirm	sim	yes
	deny	nem pensar	not a chance
	stop	já chega	enough
	inform	há 5 dias atrás	5 days ago
Chitchat	chitchat_how_are_you	tudo bem?	how are you?
	chitchat_question	tens a certeza?	are you sure?
	chitchat_bot_challenge	eu sei que não és humano	I know you're not human
	chitchat_insult	és inútil	you're useless
	chitchat_weather	como está o tempo hoje?	how's the weather today?
	chitchat_ask_name	como é que te chamas?	what's your name?
	chitchat_where_are_you	moras em Portugal?	do you live in Portugal?
	chitchat_nutrition	gostas de comida vegan?	do you like vegan food?
	chitchat_animal	tens animais de estimação?	do you have any pets?
	chitchat_random_questions_yes	gostas do teu trabalho?	do you like your job?
chitchat_random_questions_no	queres casar comigo?	will you marry me?	
Help	help_general	socorro!	help!
	help_contact	preciso de ajuda, como posso falar com os bombeiros do Barreiro?	I need help, how can I speak with the fire brigade?
	disaster_query_active	há algum incêndio em Lisboa?	is there any fire in Lisbon?
	disaster_query_past	há informação sobre cheias este mês na zona do Barreiro?	do you have any information about floods occurring this month in Barreiro?
	disaster_todo	devo sair de casa durante um terramoto?	should I leave my house during an earthquake?
	disaster_inform	há um incêndio onde eu vivo!	there's a fire where I live!
	organization_state	posso utilizar o centro de saúde do lavradio?	can I use lavradio's healthcare center?

These intents aim to provide citizens and first responders with a tool to be able to obtain critical, but generic, knowledge in disaster situations. The chitchat intents are necessary for giving the chatbot a tool to deal with user intents that are outside of its main scope, yielding a more fluid and natural conversation with its users, while also allowing for user engagement outside of disaster scenarios.

Both intents and entities are being classified using the state-of-the-art architecture Dual Intent Entity Transformer (DIET) [28]. The DIET classifier was introduced in 2020 as a multi-task transformer architecture that handles both intent classification and entity recognition together. It provides the ability to use various pre-trained embeddings, achieving competitive results with other large-scale models that are very resource and time dependent. The DIET architecture is based on a Transformer [29]—a deep learning model that is designed to handle sequential data without the need to process it in order—shared for both tasks. A sequence of entity labels is predicted by a Conditional Random Field (CRF) [30] tagging layer, on top of the transformer output sequence corresponding to the input sequence of tokens. The intent labels the transformer outputs are then embedded into a single semantic vector space. Finally, a dot-product loss is then used to maximize the similarity with the target label and minimize similarities with negative samples.

### 4.3. Dialogue Management

Managing dialogue turns is the most important aspect of yielding natural conversations. Human beings do it subconsciously by—amongst a vast range of techniques—storing key bits

of information during long chats, and implicitly expressing that information further down the conversation. For a chatbot to correctly simulate human-like conversations, several strategies are necessary.

#### 4.3.1. Slots and Forms

Slots represent part of DisBot’s memory. They act as key-value storage that can save both user-provided information, such as the name of a location, or chatbot-inferred information, such as knowledge base query results. Although this is not always true, slots tend to influence the dialogue progression. One such example is form usage. Forms are sets of slots that are required for a certain dialogue progression. They are useful because one of the most common conversation patterns is to collect pieces of information to query knowledge bases and provide an answer.

#### 4.3.2. Stories

Stories represent training examples of user conversation archetypes. They follow a specific format in which user inputs are expressed as corresponding intents and entities, while the chatbot replies are expressed as corresponding action names. They range from very simple stories, which represent only a few user intents and mapped chatbot replies, to very complex stories, which have several turns where the chatbot tries to fill slots or answer unexpected dialogue turns. DisBot has a few dozens of stories that try to simulate several possible conversations inside and outside of a disaster scenario, in order to train it for both expected and unexpected scenarios. One simple example of a story, where the user asks for the contact of an organization, is the following:

```
* greet
  - action_daytime_greet
* help_contact‘organization’: ‘Proteção Civil’
  - action_get_contact
* thanks
  - utter_no_problem
* goodbye
  - utter_goodbye
```

#### 4.3.3. Policies

Dialogue policies are responsible for deciding which action the chatbot should take in its next dialogue turn. They can be as simple as a policy that imitates stories it has been trained with, or as complex as machine learning models capable of predicting the next action based on several details, such as previous dialogue turns, and filled slots. Usually chatbots have several policies combined. DisBot uses the following policies:

**Memoization Policy** This is one of the simpler policies we use. It simply mimics the stories it has been trained on by trying to match the current five-turn fragment of the current story with the stories provided in the training data. If it finds a match, it predicts the next action of the matched story with a confidence of 1. Otherwise, it predicts none with a confidence of 0. This allows for speeding up the chatbot’s response by avoiding other policies, when possible.

**Mapping Policy** This logic allows for direct mapping between some user intents and chatbot actions. This is especially useful for when we want to add some automatic responses to users intents that will not affect dialogue progression, such as chitchat attempts.

**Form Policy** This policy is necessary for using forms. It is responsible for detecting when forms should be filled, and filling them before dialogue progression occurs, by asking the user questions about missing slots.

**Two-stage Fallback Policy** This policy allows for DisBot to fail gracefully. It handles low NLU confidence user messages—below a set intent classification threshold—by trying to disambiguate them. It asks the user to confirm the highest confidence intent and:

- If the user confirms the intent, the conversation continues as if nothing had happened;
- If the user denies the intent, the chatbot asks to rephrase the message.

When the user rephrases his message:

- If the intent classification of the rephrased message surpasses the threshold, the conversation continues;
- If not, once again, the chatbot asks the user to rephrase the message.

It is also important to mention that, in order to fail gracefully, the chatbot requires external knowledge about how to refer to each intent. In order to do this we are using a Comma-separated values (CSV) file representing, for each intent, its name and a sentence segment necessary for building the response. **TED Policy** This is our most generic and important policy, as when compared to others, it does not have a niche scenario where it is applied. The Transformer Embedding Dialogue (TED) policy [31] is used when none of the other policies are applicable, and what it does is map intents, entities, slots, active forms, and previous actions to concatenate them into a single array of features representing the last five dialogue turns. It then selects the next chatbot action by applying a dense layer to create embeddings for system actions, and calculating the similarity between the dialogue embedding and the action embeddings, based on the StarSpace algorithm [32].

#### 4.4. Response Generation

Dialogue Management is responsible for selecting DisBot’s next appropriate action, but that is not the last step of the pipeline. Response Generation is where actions occur. Actions are simply operations that the chatbot runs in response to a user’s intent. DisBot uses three types of actions.

##### 4.4.1. Utterance Actions

This is the simplest type of action. They select one out of a range of predefined answers to reply to the user. One simple example of their usage is chitchatting, where we want to have several different predefined responses for the same type of interaction, to keep conversations with our chatbot fresh and engaging. For example, when faced with the question “qual é a tua comida favorita?/what is your favorite food?”, our chatbot would randomly answer with one of the following:

- Só sei comer bits e bytes, mas de certeza que um byfinho me caía bem./I only know how to eat bits and bytes, but I would sure love a byyf;
- Adoro omeletes de bytes. Mas às vezes dão-me a volta ao sistema./I love byte omelets, but sometimes they upset my system.
- Como petisco adoro uns bons chicken-bytes./As a snack I love some good chicken-bytes;
- Nada como um belo BIToque./There’s nothing like a BIToque. (Bitoque is a Portuguese dish).

As mentioned before, the Form Policy is responsible for detecting when forms need to be filled and gathering the missing information. Utterance actions are also used as the tools to do so, having at least one possible question for each slot a form has to fill. One example, for filling the slot “time” would be:

- Pode indicar-me qual é a altura a que se refere?/Can you please tell me what time are you referring to?
- Está-se a referir a quando?/What time are you refering to?
- Em que altura?/When?

#### 4.4.2. Default Actions

Default actions are set by RASA as a means to support some policies and essential functions such as automatically stopping dialogue. These actions can be overridden as we have done with the action responsible for the Two-stage Fallback Policy.

#### 4.4.3. Custom Actions

Custom actions are where we handle the integration with our previous work, DisKnow [15]. These actions can run arbitrary code, which our chatbot uses to interact with external and internal knowledge to both build replies and fill slots and forms, to influence the dialogue flow. DisBot currently has nine distinct custom actions, which serve different purposes. They can go from very simple actions, such as replying with “Bom dia, em que o posso ajudar?/Good morning, how may I help you?” or “Boa tarde, em que o posso ajudar?/Good Afternoon, how may I help you?”, depending on the time of the day, to much more complex actions, such as inferring the knowledge base with several validations, and replying to the user according to both the extracted knowledge and information set in previous steps of the conversation.

To facilitate possible deployments of the chatbot—through image builds—and allow for multiple conversations to be held simultaneously, these actions are stored in a web server. When a policy predicts a custom action, the RASA server sends a POST request to the action server with a JSON payload including the name of the predicted action, the conversation ID, and other necessary data. The server then reacts to their call by running the code associated with the requested action and, depending on the result, optionally returns information to modify the dialogue state.

Some examples of this type are the actions that infer the knowledge graph, in which the disaster-related information is continually managed by DisKnow [15]. As mentioned before, our chatbot uses that knowledge to be able to inform citizens and first responders about disasters before, during and after their occurrence. According to the NLU training data and stories we have created, when the chatbot detects a `disaster_query_past` intent with enough confidence, it is highly probable that it will predict the next action to be a `get_past_disasters_form`, which is a custom action that uses the form policy to guarantee all required information is provided. Imagining the user provides all required information in his interaction and makes the request on the 22 of August, Figure 3 portrays how the chatbot would interact with the knowledge graph to build its reply.

As mentioned before, custom actions can also replace default actions. We have overridden the way DisBot asks the user to confirm the intent with a custom action of our own, so that the confirmation requests change depending on the intent itself, which enhances the fluidity of dialogue with the users. Figure 4 shows the process of building the first reply of the overridden default action implementing the Two-stage Fallback Policy.

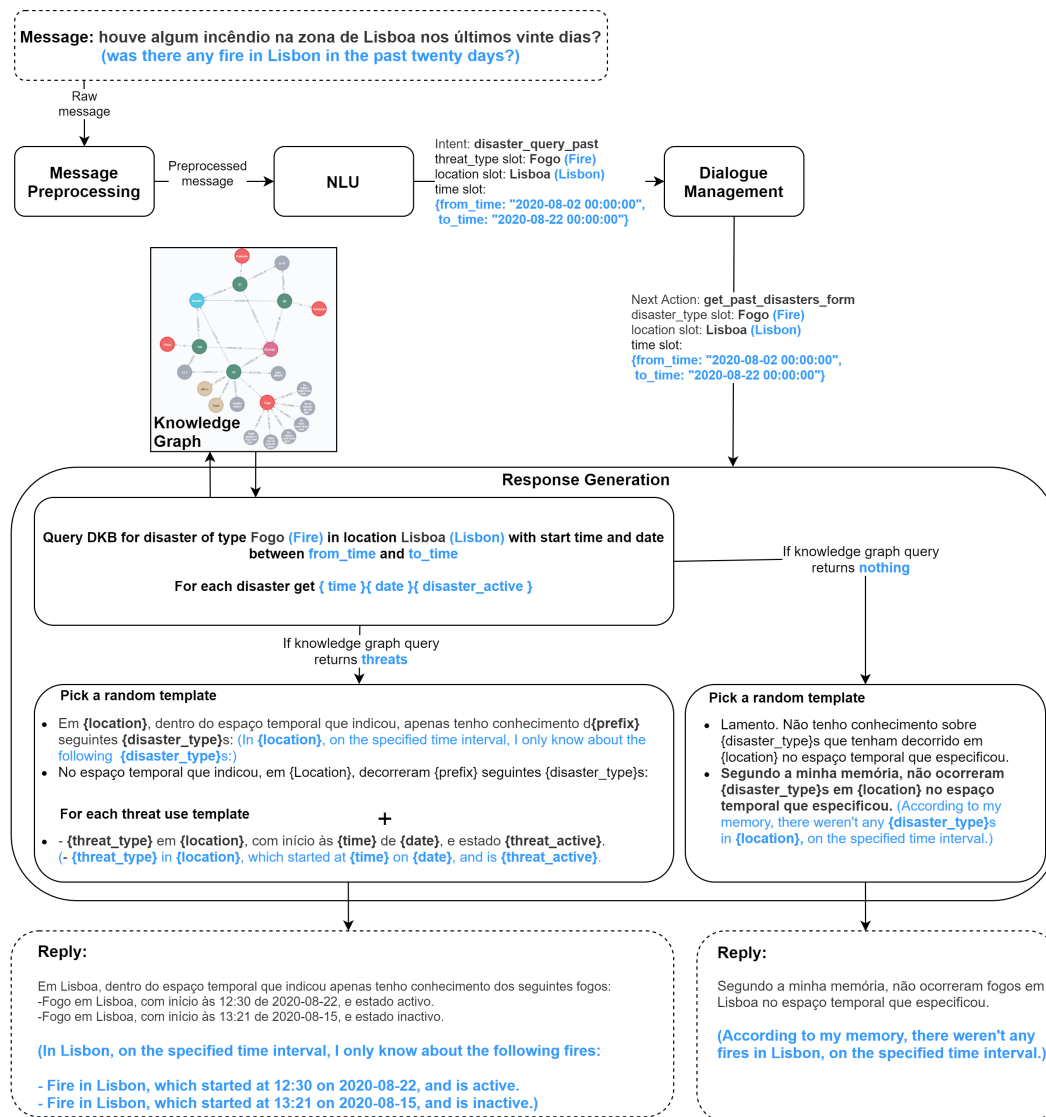


Figure 3. Chatbot response generation for the get\_past\_disasters\_form action.

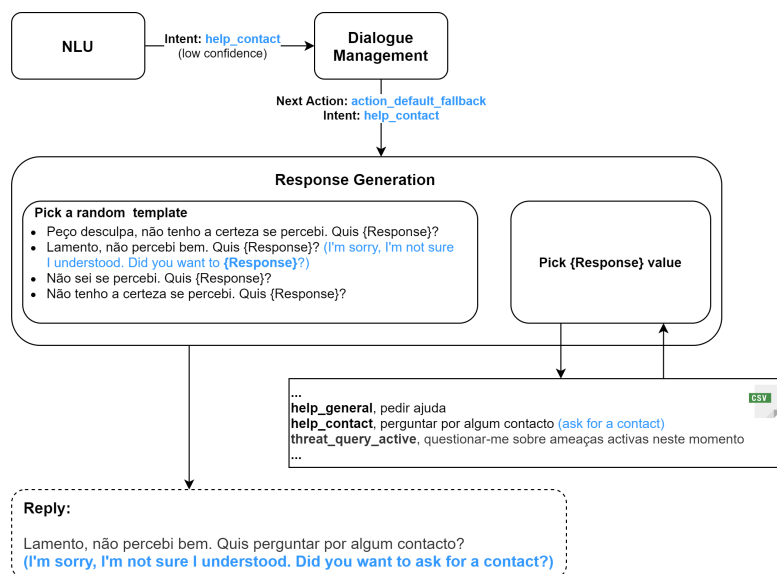


Figure 4. Building the first reply of a Fallback action.

### 5. Evaluation

As previously mentioned in the description of the methodology, the demonstration and evaluation of the created artifact was held in three different meetings. These meetings occurred in August, September, and November 2020, respectively, at the end of each iteration of the DSRM process. DisBot was presented according to a demonstration scenario, questions and answers regarding its behavior, and a small hands-on, where evaluators could test or propose interactions with the artifact.

#### 5.1. Demonstration Scenario

To demonstrate to our evaluators—the three civil protection specialists, which represent possible end-users—what our artifact does, and how well it does it, we have developed a small-scale simulated demonstration scenario, and a use-case representing user-artifact interaction during this scenario.

The scenario consists in an earthquake happening in the Lisbon metropolitan area, and being felt with high intensity in the municipality of Barreiro. The cascading effects of this earthquake lead to an explosion in a tank filled with an inflammable substance, located in one of the Sensitive Industrial Plants and Sites (SIPS) located in this municipality, FISIFE. The resulting fire then spreads to the surrounding area, also reaching Alkion, another nearby SIPS, injuring two citizens.

The use-case we have developed follows a citizen’s usage of the chatbot before, during and after the impact of these disasters. It also portrays the bulk arrival of tweets, which are processed and inserted in the Knowledge Graph by DisKnow [15], as they are posted by the citizens of the municipality that felt the earthquake, and citizens walking by in the industrial park, which have witnessed the evolution of the fire.

To further approximate the developed scenario to reality, we have requested our associates from the municipality of Barreiro to retrieve tweets related to these disasters, created by local citizens. Unrelated tweets posted by local citizens were also gathered directly from Twitter in order to represent, as close as possible, a real scenario. Figure 5 presents both the user–artifact interaction and the timeline for the demonstration use-case.

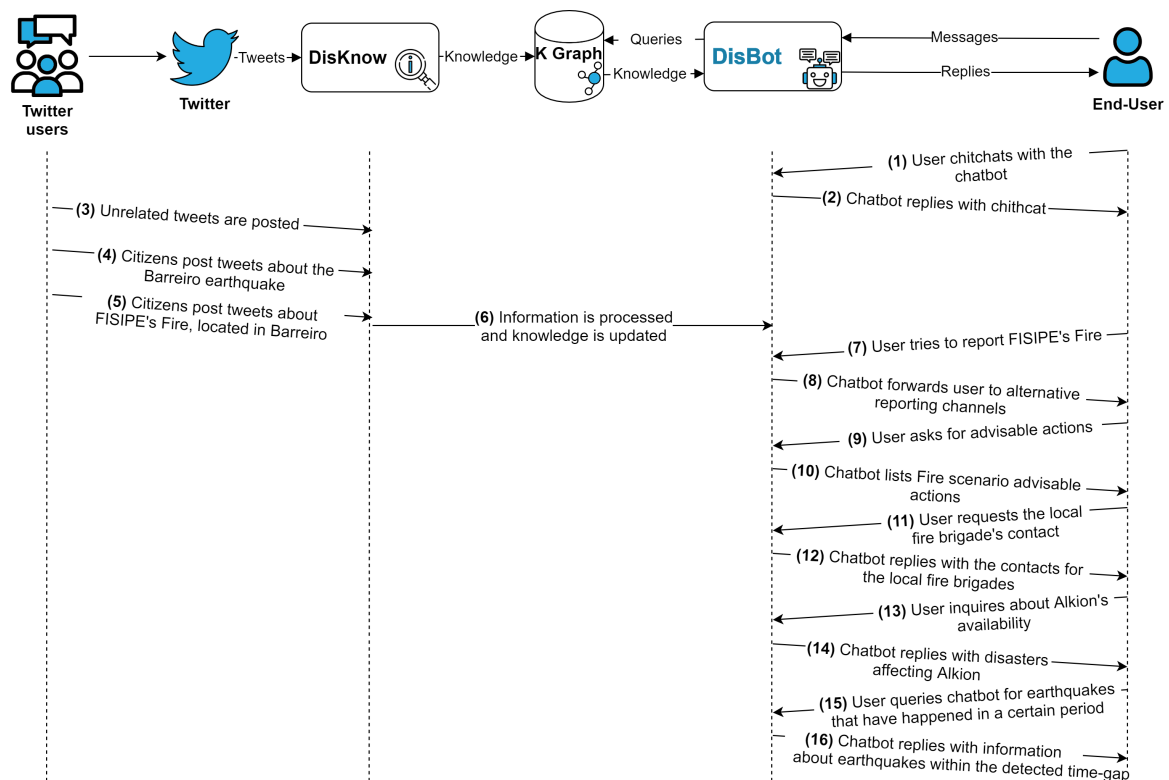


Figure 5. Demonstration use-case timeline.



### 5.2. First DSRM Iteration

The first iteration was the longest. It went from the initial identification of the problem and objectives, to the development of the first evaluated version of DisBot, which has consisted in the great majority of the developed work.

The evaluation of this iteration was performed to ensure that DisBot was fit for the purpose, according to the objective statements that were defined in collaboration with our evaluators. The obtained evaluations are presented in Table 7.

**Table 7.** Results of the first DSRM iteration.

Criteria	Objective Statement	Eval #1
Efficacy	Effectively leveraging local disaster intelligence to improve citizen and first-responder Situational Awareness and Intelligence Fusion	LA
Consistency with people/Utility	Enabling information on existing disasters, as well as advice on emergency contacts and procedures	PA
Consistency with people/Understandability	Providing fluid and understandable human-like conversations	(Third iteration)
Consistency with organization/Utility	Providing disaster management authorities with an alternative and automatic way of informing citizens	LA
Level of detail	Providing citizens with integrated local knowledge	LA
Learning capability	Automatically learning about disaster intelligence and updating that knowledge mid disaster	LA

Due to schedule limitations, for the evaluation of our first iteration only one of the three civil protection specialists was available. Despite this shortcoming, the session bore interesting results, seeing that although DisBot has been well-received by our evaluator, some shortcomings have been pointed out.

Being the first of the iterations, the demonstration did not include much local knowledge in regard to safety contacts and procedures for disaster scenarios. This has revealed to be a concern of the end-user representative (our evaluator), and has been imparted in the negative evaluation of the Consistency with people/Utility criteria.

### 5.3. Second DSRM Iteration

The second DSRM iteration was solely focused on fixing the issues that had been pointed out in the evaluation of the previous iteration, thus its starting point having been the Design and Development of our artifact. With this being the focus of this iteration, the demonstration scenario did not undergo any changes, as it was important to validate such changes in the same environment. The main developments of this iteration were the following:

- **Integration of relevant contacts of the municipality** One of the evaluators' concerns was the integration of local information in the chatbot's knowledge. This development served as a way of both deepening the chatbot's knowledge, and reassuring its possible end-users of its flexibility to integrate relevant scenario-specific local knowledge.
- **Inclusion of more training data for intent and entity recognition** Training data are never enough. The first interaction of the chatbot with its possible end-users has given us a better idea of other, until then unknown to us, possible ways of interacting with it. Adding more training data is a recurring development that allows us to improve the flexibility and naturality of the chatbot's dialogue.

Table 8 shows the obtained evaluations for this iteration.

**Table 8.** Results of the second DSRM iteration.

Criteria	Objective Statement	Eval #1	Eval #2	Eval #3
Efficacy	Effectively leveraging local disaster intelligence to improve citizen and first-responder Situational Awareness and Intelligence Fusion	FA	LA	LA
Consistency with people/Utility	Enabling information on existing disasters, as well as advice on emergency contacts and procedures	LA	LA	LA
Consistency with people/Understandability	Providing fluid and understandable human-like conversations	(Third iteration)		
Consistency with organization/Utility	Providing disaster management authorities with an alternative and automatic way of informing citizens	LA	LA	LA
Level of detail	Providing citizens with integrated local knowledge	LA	LA	LA
Learning capability	Automatically learning about disaster intelligence and updating that knowledge mid disaster	LA	LA	LA

The developments of this iteration have been well received by our evaluators—now all three of the civil protection specialists—allowing for the criteria that had previously been negatively evaluated, to be considered Largely Achieved (LA).

According to our evaluators, the main reasons as to why none of the criteria had yet reached a consensual rating of Fully Achieved (FA), were some minor grammatical errors in the chatbot replies, abrupt dialogue turns, and the necessity to include certified knowledge regarding the advice the chatbot gives for each disaster type. Finally, the evaluation criteria were discussed, and it was decided that the understandability and fluidity of the chatbot’s dialogue should be appraised in the next iteration.

#### 5.4. Third DSRM Iteration

The third and last DSRM iteration has gone back to the definition of the objectives of our artifact. In this iteration, we have included the criteria Understandability, in order to evaluate the quality of the dialogue structure, which, in this advanced stage of development, has been considered a high priority. According to these new criteria, and the last iteration concerns, the main developments of this iteration were the following:

- **Minor fixes in the dialogue system** Although minor, some issues regarding typos and hurried dialogue turns have been fixed. These included double full stops happening due to a conjunction between information in the knowledge graph and static information used the Response Generation module. Some of this module’s reply templates have also been slightly tweaked in order to enhance the fluidity between dialogue turns.
- **Integration of official disaster scenario recommendations** Official disaster scenario recommendations have been requested from Barreiro’s Municipality, and integrated in the DisBot’s knowledge, in order to emphasize the use of reliable information. To accommodate this change, and in order to deal with the high number of recommendations we have been given, the chatbot’s reply has been changed from presenting all recommendations existing for a disaster type, to presenting five random recommendations and asking if the user would like to see them all.
- **Integration of more training data for intent and entity recognition** As mentioned in the developments of the second iteration, adding more training data to the chatbot’s NLU component is a constant process of our development.

Table 9 shows the final results of the third and final iteration of DisBot’s development.

**Table 9.** Results of the third DSRM iteration.

Criteria	Objective Statement	Eval #1	Eval #2	Eval #3
Efficacy	Effectively leveraging local disaster intelligence to improve citizen and first-responder Situational Awareness and Intelligence Fusion	FA	FA	FA
Consistency with people/Utility	Enabling information on existing disasters, as well as advice on emergency contacts and procedures	FA	FA	FA
Consistency with people/Understandability	Providing fluid and understandable human-like conversations	LA	LA	LA
Consistency with organization/Utility	Providing disaster management authorities with an alternative and automatic way of informing citizens	FA	FA	FA
Level of detail	Providing citizens with integrated local knowledge	FA	FA	FA
Learning capability	Automatically learning about disaster intelligence and updating that knowledge mid disaster	LA	LA	LA

The final iteration of development has seen improved evaluations in most proposed criteria, mostly due to the minor but numerous improvements that have greatly enhanced DisBot’s interactivity. Despite not being directly relevant for its evaluation, DisBot has also been integrated in a mobile app, which helped us prove to our end-user representatives that this technology can easily be at hand for every citizen and first-responder during a disaster scenario. Despite presenting good evaluations, both the understandability and learning capability of DisBot has been considered to be the main aspects that could use some improvements before a possible deployment. The integration with other dynamic sources of knowledge can help this chatbot further evolve into a relevant disaster support tool, and improvements in the fluidity of the dialogue were also considered to be a constant development that could greatly benefit from more user interaction sessions.

### 6. Adoption Challenges

Although the evaluation results for DisBot are positive, as far as utility for crisis management is concerned, an adoption issue was identified by the evaluators, during the evaluation sessions: to ensure broad adoption and high usage of the chatbot service by the population, the chatbot user interface should be embedded into the existing municipality’s mobile app. This app is currently offered free-of-charge, is available in the Apple and Google mobile app stores, has a high adoption rate, and is used for a broad range of municipality services, including alerting functionalities. Other adoption enablers were also mentioned during the informal discussions, although not thoroughly discussed, such as integrating the crisis management functionalities into a more generic municipal digital assistant feature, and promoting chatbot usage using gamification features. Regarding adoption in less developed regions or countries, several barriers may need to be overcome, namely regarding the resilience and coverage of Information and Communications Technology (ICT) infrastructure required to support the chatbot service and the required telecommunication channels. Developing countries may also experience various barriers and challenges in the development and implementation of the solution, due to limited and low-quality of NLP datasets and pre-trained algorithms for less common languages [33].

### 7. Conclusions

The objective of this work was to devise and present an innovative chatbot artifact that uses both previous work and state-of-the-art tools to enhance citizen and first responder resilience in disaster

scenarios. The main contributions of this artifact over the current state-of-the-art were the first of its kind for the Portuguese language, and building upon the yet limited range of disaster support chatbots by managing several types of disasters.

With DisBot, we have been able to prove that chatbot technology can be used to automatically disseminate relevant knowledge to both citizens and first-responders. Through the inclusion of static local knowledge about emergency contacts, official procedures and information about organizations, our artifact has proven to be flexible when being deployed in specific disaster scenarios. This evidence has also been endorsed by specialists, through the evaluations DisBot has been given, which went from Largely Achieved to Fully Achieved on all proposed criteria. As a direct result of the interaction with these field specialists, we have proven that this novel contribution, in collaboration with disaster-related knowledge extractors, such as DisKnow [15], provides an effective disaster support system, able to independently extract and present its users with relevant resilience-improving information, and thus being a relevant tool in disaster scenarios.

Despite the good results we have achieved, DisBot has been a focus of constant development, as many user interactions bring out unexpected ways of interacting with it. Seeing that the main objective of an artifact like this is its employment in real scenarios, the interaction with specialists has also helped us understand what is expected from DisBot, and guide our development according to those expectations. We are also aware that its range of interactions is still a bit limited, still serving only as a proof of concept. DisBot could benefit from including more intents that might be relevant in disaster scenarios, and some research work will help to achieve that. One of the key issues future work needs to address is the fluidity between dialogue turns, which sometimes feels slightly unnatural and unexpected. All of these developments could be a focus of future iterations of DisBot's development, and serve as inspiration for future works in the field of disaster support chatbots.

**Author Contributions:** J.B. is a master's student and has performed all of the development work. R.R and J.C.F. are thesis supervisors and have organized all work and performed work revision. G.C is a DSRM specialist and has guided the development and validation of the work according to this methodology. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work has been partially supported by Portuguese National funds through FITEC - Programa Interface, with reference CIT "INOV - INESC Inovação - Financiamento Base", and by the Infrastrass Project—The Infrastrass project is an H2020 financed project, aiming to address cyber-physical security of Sensitive Industrial Plants and Sites (SIPS), improve resilience and protection capabilities of SIPS exposed to large scale, combined, cyber-physical threats and hazards, and guarantee continuity of operations, while mitigating effects in the infrastructure itself, the environment, and citizens in vicinity, at a reasonable cost. This work was also supported by national funds through FCT, Fundação para a Ciência e a Tecnologia, under project UIDB/50021/2020.

**Acknowledgments:** We would like to thank the European Commission for supporting this research work under the Horizon 2020 EU Framework Programme for Research (project Infrastrass, grant agreement n833088). We also thank all 27 European partners for supporting our project with their specific acknowledgements and contributions.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. Farahani, R.; Rezapour, S. *Logistics Operations and Management: Concepts and Models*; Elsevier Insights: Amsterdam, The Netherlands, 2011.
2. Bernstein, P.L. *Against the Gods: The Remarkable Story of Risk*; Wiley: Hoboken, NJ, USA, 1996.
3. Maniou, T.; Veglis, A. Employing a Chatbot for News Dissemination during Crisis: Design, Implementation and Evaluation. *Future Internet* **2020**, *12*, 109.
4. Jurafsky, D.; Martin, J.H. Speech and Language Processing, Chapter 25: Advanced Dialog Systems. In *Speech and Language Processing*; Pearson Education: Prentice Hall, NJ, USA, 2018.
5. Dale, R. The return of the chatbots. *Nat. Lang. Eng.* **2016**, *22*, 811–817.
6. Weizenbaum, J. ELIZA—A Computer Program For the Study of Natural Language Communication Between Man And Machine. *Commun. ACM* **1983**, *26*, 23–28.
7. yeung Shum, H.; dong He, X.; Li, D. From Eliza to XiaoIce: Challenges and opportunities with social chatbots. *Front. Inf. Technol. Electron. Eng.* **2018**, *19*, 10–26.

8. Belfin, R.V.; Shobana, A.J.; Manilal, M.; Mathew, A.A.; Babu, B. A Graph Based Chatbot for Cancer Patients. In Proceedings of the 2019 5th International Conference on Advanced Computing Communication Systems (ICACCS), Coimbatore, India, 15–16 March 2019; pp. 717–721.
9. Kadariya, D.; Venkataramanan, R.; Yip, H.Y.; Kalra, M.; Thirunarayan, K.; Sheth, A. kBot: Knowledge-Enabled Personalized Chatbot for Asthma Self-Management. In Proceedings of the 2019 IEEE International Conference on Smart Computing (SMARTCOMP), Washington, DC, USA, 12–15 June 2019; Volume 2019, pp. 138–143.
10. Huang, C.Y.; Yang, M.C.; Huang, C.Y.; Chen, Y.J.; Wu, M.L.; Chen, K.W. A Chatbot-supported Smart Wireless Interactive Healthcare System for Weight Control and Health Promotion. In Proceedings of the 2018 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM), Bangkok, Thailand, 16–19 December 2018; pp. 1791–1795.
11. Baby, C.J.; Khan, F.A.; Swathi, J.N. Home automation using IoT and a chatbot using natural language processing. In Proceedings of the 2017 Innovations in Power and Advanced Computing Technologies (i-PACT), Vellore, India, 21–22 April 2017; pp. 1–6.
12. D'silva, G.; Thakare, S.; More, S.; Kuriakose, J. Real world smart chatbot for customer care using a software as a service (SaaS) architecture. In Proceedings of the 2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), Palladam, India, 10–11 February 2017; pp. 658–664.
13. Choi, H.; Hamanaka, T.; Matsui, K. Design and implementation of interactive product manual system using chatbot and sensed data. In Proceedings of the 2017 IEEE 6th Global Conference on Consumer Electronics (GCCE), Nagoya, Japan, 24–27 October 2017; pp. 1–5.
14. Ranoliya, B.R.; Raghuwanshi, N.; Singh, S. Chatbot for university related FAQs. In Proceedings of the 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Udupi, India, 13–16 September 2017; pp. 1525–1530.
15. Boné, J.; Dias, M.R.; Ferreira, J.C.; Ribeiro, R. DisKnow: A Social-Driven Disaster Support Knowledge Extraction System. *Appl. Sci.* **2020**, *10*, 6083.
16. Sermet, Y.; Demir, I. An intelligent system on knowledge generation and communication about flooding. *Environ. Model. Softw.* **2018**, *108*, 51–60.
17. Tsai, M.H.; Chen, J.Y.; Kang, S.C. Ask Diana: A keyword-based chatbot system for water-related disaster management. *Water* **2019**, *11*, 1–19.
18. Chan, H.Y.; Tsai, M.H. Question-answering dialogue system for emergency operations. *Int. J. Disaster Risk Reduct.* **2019**, *41*, 101313.
19. Dias, M.; Boné, J.; Ferreira, J.; Ribeiro, R.; Maia, R. Named Entity Recognition for Sensitive Data Discovery in Portuguese. *Appl. Sci.* **2020**, *10*, 2303.
20. Peffers, K.; Tuunanen, T.; Rothenberger, M.A.; Chatterjee, S. A design science research methodology for information systems research. *J. Manag. Inf. Syst.* **2007**, *24*, 45–77.
21. Hevner, A.R.; March, S.T.; Park, J.; Ram, S. Design Science in Information Systems. *MIS Q.* **2004**, *28*, 75–105.
22. Markus, M.L.; Majchrzak, A.; Gasser, L. A design theory for systems that support emergent knowledge processes. *MIS Q. Manag. Inf. Syst.* **2002**, *26*, 179–212.
23. Cadete, G.; da Silva, M.M. Using an enterprise architecture model for assessing the resilience of critical infrastructure. Safety and Reliability-Safe Societies in a Changing World. In Proceedings of the 28th International European Safety and Reliability Conference (ESREL), Hannover, Germany, 17–21 June 2018; pp. 1459–1466.
24. Barrett, M.P. *Framework for Improving Critical Infrastructure Cybersecurity*; National Institute of Standards and Technology Tech. Rep.: Gaithersburg, MD, USA, 2018.
25. Audit, I.S.; Association, C. *Implementing the NIST: Cybersecurity Framework*; ISACA, Schaumburg, IL, USA, 2014.
26. Prat, N.; Comyn-Wattiau, I.; Akoka, J. Artifact evaluation in information systems design-science research-A holistic view. In Proceedings of the Pacific Asia Conference on Information Systems, PACIS, Chengdu, China, 24–28 June 2014.
27. International Organization for Standardization. *Information Technology—Process Assessment—Part 2: Performing an Assessment*; Standard, International Organization for Standardization: Geneva, Switzerland, 2003.
28. Bunk, T.; Varshneya, D.; Vlasov, V.; Nichol, A. DIET: Lightweight Language Understanding for Dialogue Systems. *arXiv* **2020**, arXiv:2004.09936

29. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.U.; Polosukhin, I. Attention is All you Need. In *Advances in Neural Information Processing Systems 30*; Curran Associates, Inc.: New York, NY, USA, 2017; pp. 5998–6008.
30. Lafferty, J.D.; McCallum, A.; Pereira, F.C.N. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proceedings of the Eighteenth International Conference on Machine Learning (ICML '01)*, San Francisco, CA, USA, 28 June–1 July 2001; pp. 282–289.
31. Vlasov, V.; Mosig, J.E.M.; Nichol, A. Dialogue Transformers. *arXiv* **2019**, arXiv:1910.00486.
32. Wu, L.; Fisch, A.; Chopra, S.; Adams, K.; Bordes, A.; Weston, J. StarSpace: Embed all the things! In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI)*, New Orleans, LA, USA, 2–7 February 2018; pp. 5569–5577.
33. Kshetri, N. Artificial Intelligence in Developing Countries. *IEEE Ann. Hist. Comput.* **2020**, *22*, 63–68.

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).