UNIVERSITY
INSTITUTE
OF LISBON

Construction of a disaster-support dynamic knowledge chatbot

João Miguel Baptista Boné

Master's in Integrated Business Intelligence Systems

Supervisor

Doctor Ricardo Daniel Santos Faro Marques Ribeiro, Associate Professor

ISCTE-University Institute of Lisbon

Supervisor

Doctor João Carlos Amaro Ferreira, Assistant Professor

ISCTE-University Institute of Lisbon

October, 2020

TECHNOLOGY
AND ARCHITECTURE

Construction of a disaster-support dynamic knowledge chatbot

João Miguel Baptista Boné

Master's in Integrated Business Intelligence Systems

Supervisor

Doctor Ricardo Daniel Santos Faro Marques Ribeiro, Associate Professor

ISCTE-University Institute of Lisbon

Supervisor

Doctor João Carlos Amaro Ferreira, Assistant Professor

ISCTE-University Institute of Lisbon

October, 2020

Resumo

Esta dissertação visa a conceção de um sistema de *chatbot* de apoio a desastres, com a capacidade de aumentar a resiliência dos cidadãos e socorristas nestes cenários, através da recolha e processamento de informação de fontes de *crowdsensing*, e informar os seus utilizadores com conhecimentos relevantes sobre os desastres detetados, e como lidar com eles.

Este sistema é composto por dois artefactos que interagem através de uma base de conhecimento baseada em grafos. O primeiro artefacto é um sistema de extração de conhecimento relacionado com desastres, que utiliza redes sociais como forma de explorar o conceito *humans as sensors*. Este artefacto consiste numa sequência de ferramentas de processamento de língua natural, e uma mistura de redes neuronais convolucionais e modelos baseados em léxicos, para classificar e extrair informação sobre desastres. A informação extraída é então passada para o grafo de conhecimento. O segundo artefacto, o *chatbot* de apoio a desastres, utiliza uma arquitetura *Dual Intent Entity Transformer* (DIET) para classificar as intenções dos utilizadores, e faz uso de várias políticas de diálogo para gerir as conversas, bem como armazenar informação chave. Para gerar respostas, o chatbot utiliza conhecimento local relacionado com desastres, e infere o grafo de conhecimento para extrair o conhecimento inserido pelo primeiro artefacto.

De acordo com os resultados alcançados, o nosso sistema está ao nível do estado da arte em sistemas de extração de informação sobre desastres. Ambos os artefactos foram também validados por especialistas da área, e considerados um contributo significativo na gestão de desastres.

Palavras-chave: Gestão de desastres, Processamento da Língua Natural, Inteligência Artificial, Aprendizagem Automática, *Chatbots*, Bases de Dados em Grafos

Abstract

This dissertation is aimed at devising a disaster-support chatbot system with the capacity to enhance citizens and first responders' resilience in disaster scenarios, by gathering and processing information from crowd-sensing sources, and informing its users with relevant knowledge about detected disasters, and how to deal with them.

This system is composed of two artifacts that interact via a mediator graph-structured knowledge base. Our first artifact is a crowd-sourced disaster-related knowledge extraction system, which uses social media as a means to exploit humans behaving as sensors. It consists in a pipeline of natural language processing (NLP) tools, and a mixture of convolutional neural networks (CNNs) and lexicon-based models for classifying and extracting disasters. It then outputs the extracted information to the knowledge graph (KG), for presenting connected insights. The second artifact, the disaster-support chatbot, 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, the chatbot uses local and official disaster-related knowledge, and infers the knowledge graph for dynamic knowledge extracted by the first artifact.

According to the achieved results, our devised system is on par with the state-of-the-art on Disaster Extraction systems. Both artifacts have also been validated by field specialists, who have considered them to be valuable assets in disaster-management.

Keywords: Disaster-Management, Natural Language Processing, Artificial Intelligence, Machine Learning, Chatbots, Graph Databases

Acknowledgements

I would like to acknowledge my girlfriend, parents and brother, for providing me with the emotional support and advice I needed during this journey, and most of all for putting up with me during a stressful season. Without them, this work would not have reached the point it is in today.

I would also like to deeply thank both my supervisors, Dr. Ricardo Ribeiro and Dr. João Ferreira, not only for the knowledge they have provided, but for the availability they have displayed, and indispensable support in making all of this work a reality.

Finally, I would like to thank Gonçalo Cadete, Rui Teixeira, João Branco and David Cabanas for providing us with their expertise in civil protection and DSRM, and helping us evaluate and improve our developed artifacts.

João Miguel Baptista Boné

Contents

Resumo iii

Abstract v

Acknowledgements vii

Abbreviations xi

List of Figures xiii

List of Tables xv

1 Introduction 1
1.1 Overview 2
1.2 Motivation 3
1.3 Objectives 5
1.4 The Infrastress project 6
1.5 Methodology 7
1.6 Outline of the Dissertation 13

2 Related Work 15
2.1 Disaster-related knowledge extraction systems 15
2.2 Disaster-support chatbots 19

3 Artifact 1 – Design & Development 25
3.1 Design 25
3.2 Development 26
3.2.1 Tweet Extraction and Filtering 26
3.2.2 Tweet Preprocessing 27
3.2.3 Translating 29
3.2.4 Named Entity Recognition 30
3.2.5 Text Preprocessing 31
3.2.6 Disaster Extraction 32
3.2.7 Disaster Object Generation 38

3.2.8 Knowledge Base Integration . . . 38

4 Artifact 2 – Design & Development 43

4.1 Design . . . 43

4.2 Development . . . 44

4.2.1 Message Preprocessing . . . 45

4.2.2 Natural Language Understanding . . . 45

4.2.3 Dialogue Management . . . 48

4.2.4 Response Generation . . . 50

5 Demonstration & Evaluation 55

5.1 Demonstration Scenario . . . 55

5.2 Evaluation . . . 58

5.2.1 First DSRM Iteration . . . 58

5.2.2 Second DSRM Iteration . . . 60

5.2.3 Third DSRM Iteration . . . 62

6 Conclusions & Future Work 67

6.1 Conclusions . . . 67

6.2 Communication . . . 69

6.3 Future Work . . . 70

References 70

A Related Tweets 77

B Unrelated Tweets 79

C Chatbot Interaction 81

Abbreviations

AI	**A**rtifical **I**ntelligence
API	**A**pplication **P**rogramming **I**nterface
BIM	**B**uilding **I**nformation **M**odeling
CDD	**C**onversation **D**riven **D**evelopment
CNN	**C**onvolutional **N**eural **N**etwork
COBIT	**C**ontrol **Ob**jectives for **I**nformation and related **T**echnologies
CSV	**C**omma **S**eparated **V**alues
DIET	**D**ual **I**ntent **E**ntity **T**ransformer
DKB	**D**ynamic **K**nowledge **B**ase
DSRM	**D**esign **S**cience **R**esearch **M**ethodology
GNMT	**G**oogle **N**eural **M**achine **T**ranslation
IF	**I**ntelligence **F**usion
KPI	**K**ey **P**erformance **I**ndicator
LSTM	**L**ong **S**hort **T**erm **M**emory
MDPI	**M**ultidisciplinary **D**igital **P**ublishing **I**nstitute
NER	**N**amed **E**ntity **R**ecognition
NLP	**N**atural **L**anguage **P**rocessing
NLTK	**N**atural **L**anguage **T**ool **K**it
NLU	**N**atural **L**anguage **U**nderstanding
PBMT	**P**hrase **B**ased **M**achine **T**ranslation
RAF	**R**esilience **A**ssessment **F**ramework
SA	**S**ituational **A**wareness

SIPS	**S**ensitive **I**ndustrial **P**lants and **S**ites
TED	**T**ransformer **E**mbedding **D**ialogue

List of Figures

1.1 High-level framework of the artifacts' interaction 5

1.2 Dissertation scope inside of the Infrastress Portuguese context 7

1.3 DSRM process model 8

1.4 Disaster risk management resilience assessment framework 10

1.5 Hierarchy of criteria for IS artifact evaluation 11

3.1 Artifact 1 - Disaster extraction framework 26

3.2 Tweet extraction scheme 27

3.3 Tweet preprocessing example 29

3.4 Named Entity Recognition using Local and SpaCy Knowledge 31

3.5 Disaster extraction component pipeline 33

3.6 Disaster type classification CNN 34

3.7 Loss evolution through the training epochs of the first CNN (Related Classification) 35

3.8 Loss evolution through the training epochs of the second CNN (Type Classification) 35

3.9 Yoon Kim model architecture with two channels 36

3.10 Simplified Knowledge Base integration flow chart 40

4.1 Artifact 2 - Chatbot framework 44

4.2 Schematic representation of the DIET architecture 46

4.3 Building the first reply of a Fallback action 53

4.4 Chatbot response generation for the `get_past_disasters_form` action . . 54

5.1 Demonstration use-case timeline 57

5.2 DSRM Iterative cycles scheme 58

5.3 Snippet of the Knowledge Base after tweet ingestion 64

List of Tables

1.1 Objective statements to be used in the DSRM evaluation 12

2.1 Filtering criteria used for the knowledge extraction systematic review . . . 16
2.2 Study selection for the knowledge extraction systematic review 16
2.3 Filtering criteria for the disaster-support chatbots systematic review 21
2.4 Study selection for the disaster-support chatbots systematic review 21
2.5 Summary of the systems included in the systematic review 24

3.1 Comparative results of the event relatedness classification task 37
3.2 Comparative results of the event type classification task 38

4.1 Used featurizers and related functions 45
4.2 Chatbot implemented intents with unannotated examples 47

5.1 Results of the first DSRM iteration 59
5.2 Results of the second DSRM iteration 61
5.3 Knowledge Extraction accuracy according to tweet predetermined class . . 63
5.4 Estimated results of the third DSRM iteration 65

Chapter 1

Introduction

Several types of human-made and natural disasters are continually challenging our society. These scenarios are extremely hard to predict and prevent, but they can be handled, improving community resilience in the face of these adversities.

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 been seeing 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 an uprising in disaster management technologies. However, the use of social crowd-sourcing data and information dissemination technologies, like chatbots, is still very lacking in this field of study, although their potential for improving authorities' decision making and population access to information, is tremendous.

Information extraction techniques are, nowadays, one of the most relevant topics in disaster management, and many researchers have examined these approaches in order to establish pre and post-disaster response programs [3]. Its potential, however does not come without its challenges. There is a huge scarcity of formalized data, such as domain-specific annotated data sets, as well as severe problems regarding the use of social media data, such as the lack of methods to filter trustful information and the challenge that is exploiting noisy social media data streams of high volume, in real-time [4]. Setting aside the challenges of extracting it, this information also needs to be thoroughly disseminated to the population and relevant authorities, and thus, there is also a growing need for reliable communicators. Chatbots, also commonly called digital assistants, can fill this role in an automatic, centralized and inexpensive way.

Chatbots are machine agents that serve as natural language user interfaces for data and service providers [5], and they have been around for a while as both personal assistants and information facilitators. We have seen 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 key words. Chatbots have since come a long way, from Parry in the 70s to the focus in task-completion virtual assistants in the 2000s, followed by the big companies' personal assistants, like Alexa and Siri, and the recent social chatbots, such as Microsoft's XiaoIce [7].

1.1 Overview

This dissertation aims to explore, plan, and create an end-to-end disaster-support system. The system was jointly developed at Iscte - Instituto Universitário de Lisboa and INOV-INESC Inovação, and is part of Infrastress, a project financed by the Horizon 2020 (H2020) European Union funding for Research & Innovation.

Our system makes use of state-of-the-art Data Extraction and Natural Language Processing (NLP) techniques to provide both population and relevant authorities with pertinent real-time information, gathered using crowd-sensing processes and disseminated using a chatbot. It will both use existing pre-trained models, selected through a comparison with different methods, as well as present a new state-of-the-art model for extraction of disasters from social media posts, and the creation of a dynamic graph knowledge base, from architecture to implementation.

It is also worth mentioning that the proposed work is adaptable for different scenarios and languages, through the use of translation techniques and easily augmented pipelines for relevant data extraction.

1.2 Motivation

Throughout the world, earthquakes, floods, fires and other natural hazards cause tens of thousands of deaths and billions of euros in economic losses each year [8]. It goes without saying that these numbers only rise, if we take into consideration other risk-inducing disasters, like terrorism or even pandemics, as we have recently seen with the spread of COVID-19.

Advances in information technology and communications, combined with the introduction and uprising of social media apps, have created a new world of emergency and disaster management services by allowing impacted people to produce real-time geo-referenced information on critical incidents [9]. These media have been playing a huge role in disaster management, however, although they are used to communicate emergency information and urgent requests between authorities and those affected by disasters [10], this flow of information is still lacking. A huge slice of relevant information is still being posted on these media, without a connection to the real stakeholders of said disasters, and thus, no real effect. One of the challenges of being able to use this information is filtering it, considering that social media posts tend to vary widely in both their subjects and utility, ranging from off-topic to relevant disaster-related information. Gathering actionable information can improve disaster response and greatly reduce both property and human casualties [11].

It is also extremely important to understand that extracting information is not the only step in the road to improve communication in disaster scenarios. It is critical to be able to efficiently provide that same information to both relevant authorities and population.

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 [12]. Chatbots 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. 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. These entities have, nevertheless, the tremendously challenging task of working efficiently in open domain scenarios [7]. In closed domain scenarios, however, 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 focused solely in decision makers, and formal knowledge such as official reports, for a very limited amount of disasters. The possibility of creating a system that takes advantage of the large amounts of data human beings are constantly creating, and is able to summarize it and inform disaster stakeholders in an automatic and reliable way would greatly improve human resiliance 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 this system's value and contribute to disaster management.

1.3 Objectives

Taking into account the identified problems, the research performed on this dissertation aims to devise a disaster-support chatbot system with the capacity to gather and process information from crowd-sensing sources and to inform its users with knowledge about detected disasters and how to better deal with them. With this objective in mind, the main intent of our work is to create the following two innovative artifacts:

1. A crowd-sourced disaster-related knowledge extraction system;
2. A disaster-support chatbot.

Our research has two main research question:

1. *Can enough relevant information be automatically crowd-sourced in disaster scenarios?*
2. *Can a chatbot improve citizens and first responders' informativeness in a disaster scenario?*

The two proposed artifacts are independent, but integrated to function as a whole, in what we consider to be a disaster-support dynamic knowledge chatbot, a system capable of both extracting and communicating knowledge in real-time to its users, improving their resilience in disaster scenarios. Figure 1.1 lays out a high-level pipeline on how these two artifacts are connected inside of the proposed disaster-support system.

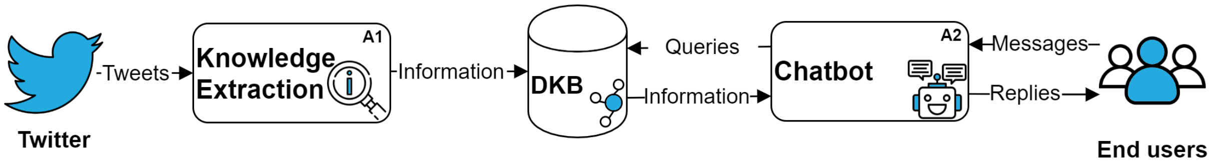

FIGURE 1.1: High-level framework of the artifacts' interaction

1.4 The Infrastress project

As mentioned before, Infrastress is an H2020 financed project. This project has the participation of 27 European partners with complementary roles and skills. Its main objective is to address cyber-physical security of Sensitive Industrial Plants and Sites (SIPS), and 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 reasonable cost.

This dissertation's devised system is part of a component of Infrastress developed at INOV-INESC Inovação, and also uses one of its pilots and several contributors to validate its usage in a simulated real-world environment.

The Portuguese Infrastress implementation consists of a system that aims to mitigate the risks citizens on the vicinity of disasters are exposed to. This is done through a mobile app that integrates functionalities such as reporting a disaster occurrence, getting informed about new and occurring disasters, and also chatting with a digital assistant about such disasters and how to better deal with them. Occurrences can be reported in two different ways: through a form in the app, or directly in the Building Information Modeling (BIM) model. The last one is a three-dimensional model of the vicinity, where a user can navigate, access, and register information.

Although these are the only two ways for a user to directly register new disasters, the system is constantly scouting for new disasters through both local tweets and messages exchanged in the Infrastress app's integrated chat.

On top of this user visualization, there are also some admin-specific functionalities, such as monitoring, through graphs and maps, the civil opinion on how the local authorities are dealing with each active disaster. This information consists in the geolocation placement of each opinion, and its sentiment analysis.

All of these components are interconnected through a central message distribution hub. It consists in an Apache Pulsar[1] implementation, an open-source distributed pub/sub messaging program that was originally developed at Yahoo[2]. All components of the system can be consumers and/or producers of several topics, and when a message is produced by a component in a certain topic, it is disseminated for all components who are consumers on that same topic. Each component then decides whether to consume or discard that information, allowing for each component to be internally independent of each other.

Although the European scope of the project is wider, the work developed for this dissertation focuses only on the Portuguese extent of the project. Figure 1.2 specifies the dissertation scope inside of the Portuguese context of Infrastress.

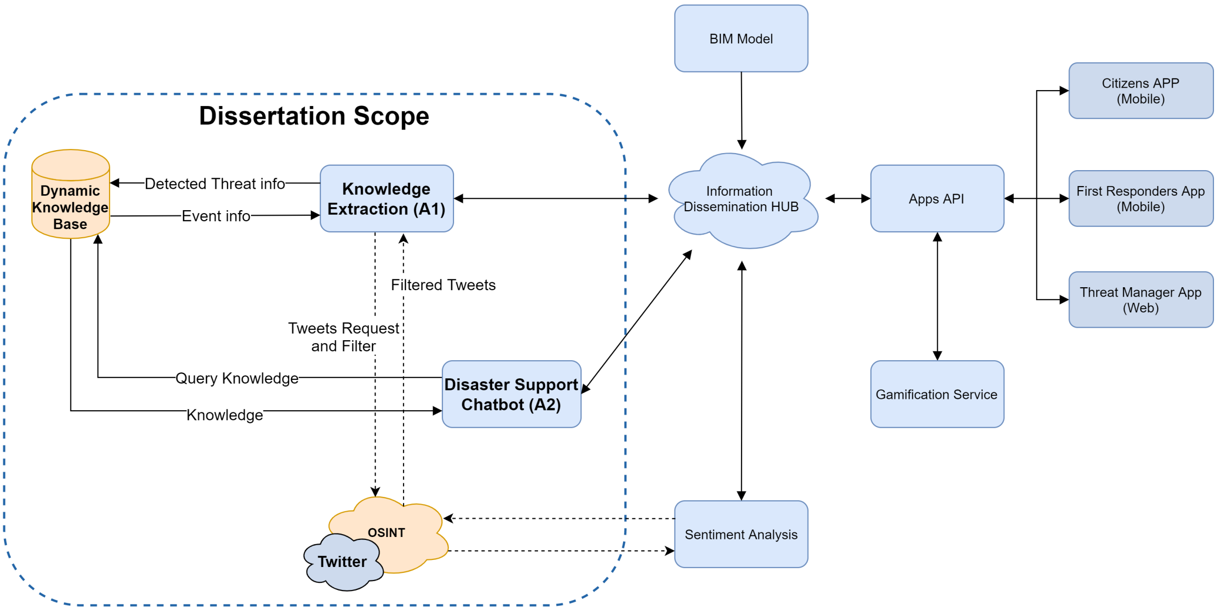

FIGURE 1.2: Dissertation scope inside of the Infrastress Portuguese context

1.5 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

---

[1] http://pulsar.apache.org/
[2] https://www.yahoo.com/

of significant importance [13]. 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 [14].

Taking this into account, this dissertation follows the Design Science Research Methodology (DSRM), along with the seven guidelines proposed by Hevner et al. [14]. 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. Figure 1.3 pertains to a nominal sequence consisting of six activities that, according to its authors, resumes the DSRM process.

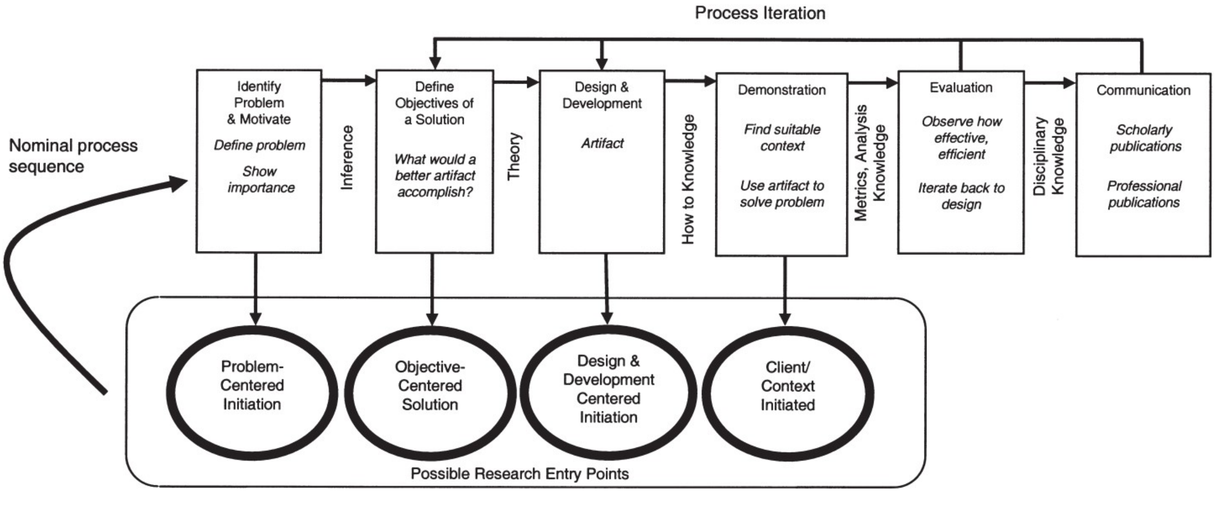
Process Iteration
Nominal process sequence
Identify Problem & Motivate
Define problem
Show importance
Inference
Define Objectives of a Solution
What would a better artifact accomplish?
Theory
Design & Development
Artifact
How to Knowledge
Demonstration
Find suitable context
Use artifact to solve problem
Metrics, Analysis Knowledge
Evaluation
Observe how effective, efficient
Iterate back to design
Disciplinary Knowledge
Communication
Scholarly publications
Professional publications
Problem-Centered Initiation
Objective-Centered Solution
Design & Development Centered Initiation
Client/ Context Initiated
Possible Research Entry Points

Figure 1.3: DSRM process model, Peffers et al. [13]

There are four different entry points – also known as approaches – to the DSRM, but the one we have used is the first one, Problem-Centered-Initiation, as it is by its own nature, the starting point of our system and the Infrastress project itself.

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 [15].

Our work consists in three iterations of the DSRM process. The first iteration incorporates the majority of this dissertation's work, going from the initial entry point, to the design and development (Chapters 3 and 4) of the first version of the artifacts, as well as their evaluation (Chapter 5). As this methodology aims to be interactive as the development goes on, the entry point of the second and third iterations of the process are only decided during the conclusion and evaluation of their previous iteration.

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 Rui Teixeira, João Branco and David Cabanas, three civil protection specialists from the Portuguese municipality of Barreiro, alongside with Gonçalo Cadete, a certified specialist in COBIT5 (Control Objectives for Information and Related Technologies) auditing, and experienced DSRM researcher. These three civil protection specialists were also the evaluators and stakeholder representatives for all three iterations of the DSRM process.

To objectively define the DSRM evaluation criteria, which are a drill-down of the dissertation objectives that we have previously defined, we first need to determine our artifact's capability. 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 [16]. The framework is presented in Figure 1.4.

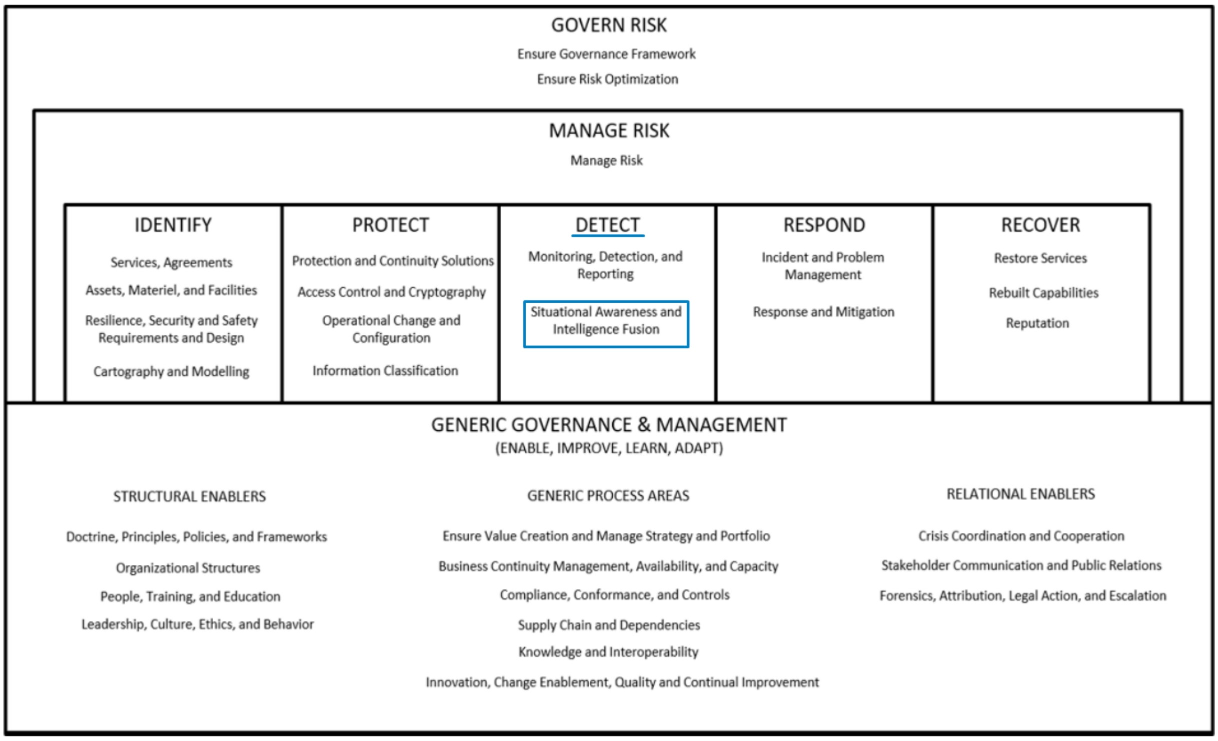
GOVERN RISK
Ensure Governance Framework
Ensure Risk Optimization
MANAGE RISK
Manage Risk
IDENTIFY
Services, Agreements
Assets, Materiel, and Facilities
Resilience, Security and Safety Requirements and Design
Cartography and Modelling
PROTECT
Protection and Continuity Solutions
Access Control and Cryptography
Operational Change and Configuration
Information Classification
DETECT
Monitoring, Detection, and Reporting
Situational Awareness and Intelligence Fusion
RESPOND
Incident and Problem Management
Response and Mitigation
RECOVER
Restore Services
Rebuilt Capabilities
Reputation
GENERIC GOVERNANCE & MANAGEMENT
(ENABLE, IMPROVE, LEARN, ADAPT)
STRUCTURAL ENABLERS
Doctrine, Principles, Policies, and Frameworks
Organizational Structures
People, Training, and Education
Leadership, Culture, Ethics, and Behavior
GENERIC PROCESS AREAS
Ensure Value Creation and Manage Strategy and Portfolio
Business Continuity Management, Availability, and Capacity
Compliance, Conformance, and Controls
Supply Chain and Dependencies
Knowledge and Interoperability
Innovation, Change Enablement, Quality and Continual Improvement
RELATIONAL ENABLERS
Crisis Coordination and Cooperation
Stakeholder Communication and Public Relations
Forensics, Attribution, Legal Action, and Escalation

Figure 1.4: Disaster risk management resilience assessment framework, Cadete and da Silva [16]

Our main objective, with this work, is to devise a disaster-support chatbot system with the capacity to gather and process information from crowd-sensing sources and to inform its users with knowledge about detected disasters and how to better deal with them. Crossing that objective with the RAF in Figure 1.4, 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 an homogeneous idea, artifact evaluation is still discussed among this methodology's community, as in the DSRM literature, evaluation criteria are presented in a fragmented or incomplete manner [17]. To overcome this obstacle, we have decided to follow the hierarchical evaluation criteria for IS artifacts proposed by Prat et al. [17]. Figure 1.5 presents this hierarchy, along with the highlighted criteria we evaluate our artifacts with.

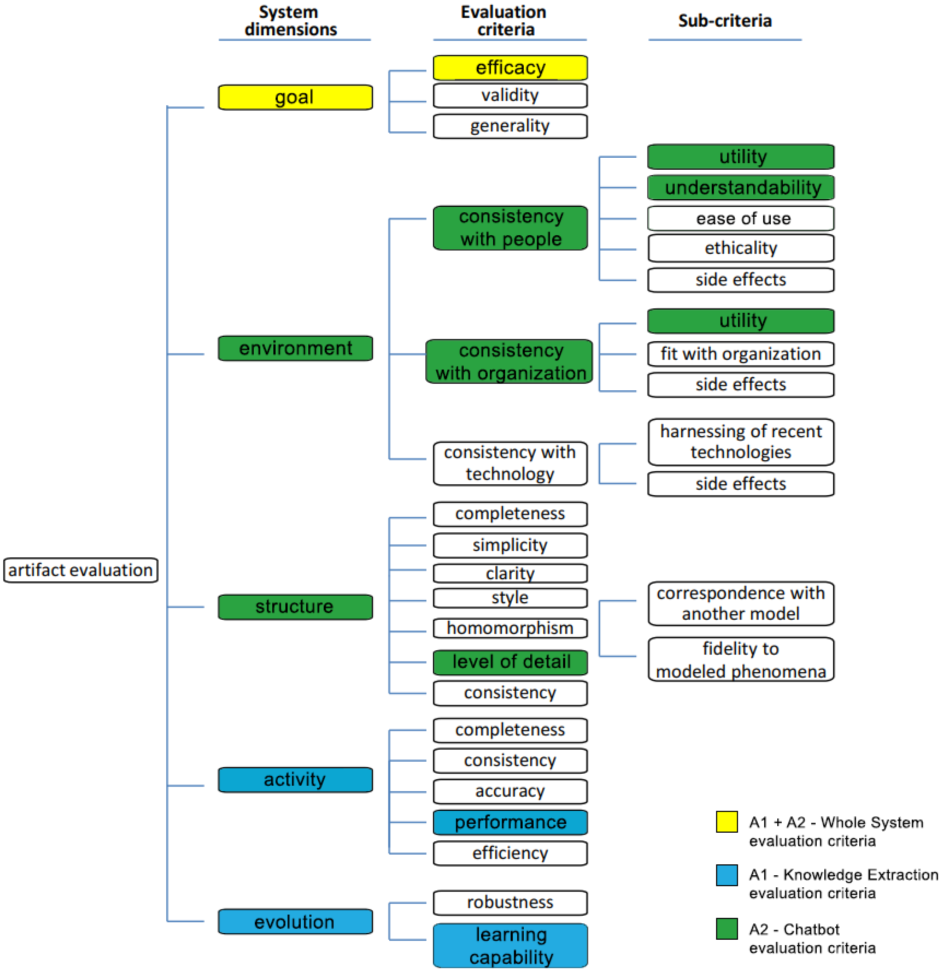

FIGURE 1.5: Hierarchy of criteria for IS artifact evaluation, Prat et al. [17]

According to both the selected capability and criteria, Table 1.1 displays the created objective statements, which serve as objectives to be evaluated in the three iterations of the DSRM process (Chapter 5).

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 [1]
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	Activity	Performance	Correctly extracting disasters and related information amongst noisy social media environments
SA & IF	Evolution	Learning capability	Automatically learning about disaster intelligence and updating that knowledge mid disaster

TABLE 1.1: Objective statements to be used in the DSRM evaluation

Ratings are given by each evaluator based on evidence showing that the objective statement's added value has been fulfilled. With this in mind, we have decided to use ISO 15504's four-point NLPF scale [18], which consists in the following four levels:

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

---

[1] Although presented here, this criteria has only been added on the 3rd iteration.

1.6 Outline of the Dissertation

Having its objectives and methodology outlined, the structure of this dissertation is composed of six chapters, including the Introduction (Chapter 1). It comprises of the following structure:

Chapter 2 presents systematic reviews on the state-of-the-art of both disaster-related knowledge extraction systems and chatbot systems applied to disaster-management, respectively.

Chapter 3 provides a high-level view of the system's first artifact, the knowledge extraction system, and its architecture. This chapter also focuses on the hands-on development of the first DSRM iteration, giving a detailed insight on every component of the knowledge extraction system, as well as why they were chosen, how they were implemented, and how they add value to our system as whole.

Chapter 4 provides a high-level view of the system's second artifact, the disaster-support chatbot, and its architecture. This chapter also focuses on the hands-on development of the first DSRM iteration, giving a detailed insight on every component of the chatbot, as well as why they were chosen, how they were implemented, and how they add value to our system as whole.

Chapters 5 outlines the scenario in which our artifacts were tested. This includes specification about the environment, participants, and storytelling. Also on this chapter, the evaluation of our artifacts is presented, observing the results of the criteria we have set to evaluate, for each of the three iterations of the DSRM process.

Chapter 6 provides a summary of the work we have developed. This chapter also highlights the contributions of our work and exposes its limitations, with a clear focus on improvement, and thus, arriving at what we believe to be the future work to be held.

Chapter 2

Related Work

2.1 Disaster-related knowledge extraction systems

A disaster can be defined as a source of danger, and its consequences can adversely affect the world in different ways. Digital Social networks have both emerged and evolved greatly in the 2000s, and the large amount of information shared can be of great help in disaster management and in other critical situations [19].

We have led a systematic review using Scopus[1] as our primary research database, and both ACM [2] and Google Scholar[3] as secondary sources for papers, the last one requiring a more restrictive approach, due to the range of quality of the works it contains. With this in mind, we have built a research query for the extraction and filtering of works from these databases. This query was the following:

(("disaster extraction" OR "disaster classification" OR "crisis extraction" OR "crisis classification") AND ("social network" OR "social media" OR twitter OR crowdsensing OR "crowd sensing"))

---

[1] https://www.scopus.com/home.uri

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

[3] https://scholar.google.com/

The search string inquiry returned 26,156 studies. Due to their quantity, and quality of the filtered results, we have decided not to make any changes to our research query. The extracted works were filtered according to three phases: inclusion and exclusion criteria, abstract screening, and full-text screening. We have also set a number of criteria for the first of these filtering stages. These criteria refer to the content of the works, their form of publishing, and their availability. Table 2.1 displays all of them.

Filtering Criteria	
Inclusion criteria	**Exclusion criteria**
Written in English or Portuguese	Not written in English nor Portuguese
Publication date after/during 2005	Publication date before 2005
Scientific papers in conferences or journals	Non-free documents outside of ISCTE's license or documents published uniquely in non-reliable sources
Title relevance in respect of the extraction of disaster-related information from social media or other crowd-sensing sources	No title relevance in respect of the extraction of disaster-related information from social media or other crowd-sensing source

Table 2.1: Filtering criteria used for the knowledge extraction systematic review

Table 2.2 shows the results of all filtering stages, numbering the selected studies for each of them.

Filtering phase	**Number of studies**
Search string inquiry	26,156
Inclusion and exclusion criteria	162
Abstract screening	24
Full-text screening	14

Table 2.2: Study selection for the knowledge extraction systematic review

Major systematization work on Twitter usage as an information source was carried out by Steiger et al. [20], where they highlight that 46% of papers were about event detection. Furthermore, they report that 71% of the reviewed studies have no specific application context and are mainly focused on creating system architectures. This led the authors to the conclusion that disaster management is an opportunity to apply such knowledge to, and could significantly benefit from it, in order to strengthen situation awareness and improve emergency response. It shows that event detection on Twitter is a popular domain and plays an essential role in Twitter research, which is done based on a keyword search directly over Twitter's Application Programming Interface (API). It is also common to use control vocabularies as a means to filter which tweets will be captured and retrieved through the API. Examples of these approaches are the keyword-based filtering to identify predefined events like forest fires [21], earthquakes [22] and floods [23]. Although keyword-based data extraction might be useful, its standalone usage has several associated problems, like noise and bias. There is also a high rate of false results associated with these methods, such as detecting that there is going to be a fire in New York from the tweet "Arcade Fire are going to be playing in NYC". An example of these problems is shown by the work of Bosley et al. [24], where they identify that only 25% of the tweets generated by their initial keyword search were related to their topic. In 2013, through a more detailed analysis of the #TFF hashtag for the Tobacco Free Florida media campaign, the authors found that out of 3104 selected tweets, only 1% were actually related to it [25]. So, the need for methods that can go beyond keyword filtering has been, and still is, one of the main focuses of Twitter research efforts to date.

The research literature on social networking and social media applied to disasters and crisis situations is still quite limited. Some researchers, however, have examined these approaches in order to establish pre and post-disaster response programs [3]. One example would be the work of Sakaki et al. [26] where Twitter was used to construct an earthquake reporting system in Japan, with an extremely positive result of 96% of said earthquakes over Richter' magnitude 3 being detected, directly through monitoring tweets. In another example use-case, demonstrating that collecting trustworthy information is crucial for disaster management, Uddin et al. [27] used Apollo [28] to collect and evaluate tens of thousands of tweets about New York City's gas availability during and shortly after 2012's Hurricane Sandy to determine the accuracy of individual tweets, while considering unknown source reliability and uncertain provenance. Their results have shown that 90%

of the tweets classified as veracious by Apollo were true, which is very relevant given that less than half of the tweets were actually true. CrisisTracker [29] has also proven to be a great example of using social media in disaster management, and probably one of the most successful ones. Although it has been used to monitor events like Fukushima's nuclear disaster, its largest field trial dealt with the 2012 civil war in Syria. During the 8-day study, CrisisTracker processed an average of 446,000 tweets a day and managed to successfully reduce the information to consumable stories, being successful in raising the situational awareness of such disaster areas. In practice, it took about 30 minutes after an isolated incident occurred before CrisisTracker could reduce the overload of social media information to a consumable story.

Disaster classification can be seen as a very specific application of event detection. The latter is a task whose purpose is to identify related stories from a continuous stream of textual data, and considers an event as being something that occurs at a certain time in a certain location. In 2015, Atefeh and Khreich [30] have surveyed the methods used in event detection from social media, and have reported the usage of various machine learning techniques, such as classifiers, clustering and language models. Recently, however, deep learning has emerged on most Natural Language Processing (NLP) tasks, and event detection was not an exception. Several works have introduced the application of deep learning techniques to address event detection at a sentence level. This has been done by first identifying event triggers – which, in a sentence can be, for example, verbs or derivational nominalizations – and then classifying them into specific types.

Deep learning models have proven to be more tolerant to domain and context variations, when compared with keyword-based models. One of the reasons being that the usage of word embeddings – a technique where words or phrases are mapped to vectors in an n dimensional space – allows for a more generic and richer representation of words and their relationship [31]. There is not, however, much work in regard to disaster classification in tweets using machine learning, as this is generally done using keyword-based approaches, and labeled tweets for this task are scarce. Nevertheless, a few works have been published in recent years, showing promising results.

In 2017, Burel et al. [32] have contributed to the field with the use of a type of neural networks – computing systems simulating a circuit of neurons – called dual convolutional neural network (Dual-CNN). This network was used to both see if tweets were, or not, related to a disaster, and which type of disaster they were related to. This Dual-CNN has an additional semantic representation layer which vectorizes information regarding the tweet's detected named entities. The authors have reported very little difference between applying Dual-CNN and a single CNN, for this specific task. Despite the semantic knowledge used by the first, its results have not shifted, and in some tasks it has even performed worse. More recently, in 2019, Sit et al. [33] have applied long short-term memory (LSTM) neural networks to the disaster relatedness task, while comparing its results with linear machine learning algorithms, like support vector machines (SVM) and logistic regression. Although the results have not improved over [32], the deep learning approaches have clearly improved on the results achieved by both SVM and logistic regression models. Also in 2019, Alam et al. [34] have used several models for the disaster type classification task, ranging from traditional machine learning models, such as naïve Bayes classifier, random forests and SVM, to a deep learning CNN. Although the CNN has outperformed most models, the SVM has shown results on par with it, coming up to an accuracy of 93%.

2.2 Disaster-support chatbots

Chatbots have enormous potential as knowledge disseminators in time-critical scenarios, like 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[4] as our primary research database, and Google Scholar[5] as a secondary source for papers, which required a more restrictive

---

[4] https://www.scopus.com/home.uri

[5] https://scholar.google.com/

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 through 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: inclusion and exclusion criteria, abstract screening, and full-text screening. For the first of these filtering phases we have set an assortment of criteria, related to the works' content, their type of publication, and their availability. Table 2.3 displays all of these criteria.

Filtering Criteria	
Inclusion criteria	**Exclusion criteria**
Written in English or Portuguese	Not written in English nor Portuguese
Applied to the English language	Applied to any other languages besides English
Publication date after/during 2010	Publication date before 2010
Scientific papers in conferences or journals	Non-free documents outside of ISCTE's license or documents published uniquely in non-reliable sources
Title relevance in respect of chatbots and crisis-support	No title relevance in respect of chatbots and crisis-support

Table 2.3: Filtering criteria for the disaster-support chatbots systematic review

Table 2.4 shows the results of all filtering stages, numbering the selected studies for each of them.

Filtering phase	**Number of studies**
Search string inquiry	163
Inclusion and exclusion criteria	23
Abstract screening	9
Full-text screening	3

Table 2.4: Study selection for the disaster-support chatbots systematic review

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 [35] have 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 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. Its 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 preassigned weights and aim to provide the system with information regarding which databases to use, which analyses to apply, and in which format should the answer be provided with. 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. [36] have proposed a water-related disaster support chatbot system named Ask Diana, which has ties with Taiwan's Water Resource Agency. This system's purpose 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 catalogued 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 key-words that the system then matches with a static built-in mapping table, as to understand the user's intent. Afterwards 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 system's labor-intensive maintenance, as the keyword mapping table is handcrafted and already contains up to 200 keywords, making further updates very time-consuming.

Also in 2019, the system by Chan and Tsai [37] was led by one of the authors of [36]. 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 question's class and then proceeds to using 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 path's formation, they do refer that all distances between edges were set to 1. 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 through 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 2.5 presents a summary, and comparison, of these chatbot systems.

Paper	**Year**	**Disasters**	**Knowledge**	**Natural Language Understanding**	**Knowledge Lookup**	**Replies**
Tsai et al. [36]	2019	Floods, Droughts	Pre-classified static files database	Keyword-based mapping table	Fuzzy search decision tree	Images
Sermet and Demir [35]	2018	Floods	API to a web-based platform	Ontology-based extraction	Question models with pre-assigned weights	Images, Natural Language
Chan and Tsai [37]	2019	Floods, Droughts	MongoDB with static and dynamic information	Ontology-based extraction	Path planning and query formation	Natural Language

Table 2.5: Summary of the systems included in the systematic review

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 [36] has limited their system's possible interactions, 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 [37] and [35], as disaster management requires both real-time updated data about disasters taking place, and predefined information on how to deal with specific scenarios.

It is also relevant to understand that other types of disasters, besides floods and droughts, can also provide very interesting and challenging scenarios to apply chatbots to. And even though existing efforts are aimed towards providing information to relevant agencies and decision-makers, there is an opportunity at exploring the usage of chatbots providing information to a disaster's nearby population and first-responders.

Chapter 3

Artifact 1 – Design & Development

3.1 Design

Our first artifact is the knowledge extraction system. This artifact is responsible for the first half of our diaster-support system. It collects and filters tweets directly from Twitter, and then, both processes and translates their texts to English. This artifact is also responsible for the extraction of knowledge, meaning it needs to classify if a tweet is related to a certain type of disaster, and to extract all additional information that helps our system build the knowledge represented in the knowledge base. Its last step consists in the association of all extracted data, to create disaster objects that can be represented in the knowledge base, as well as all of the validations required to guarantee that our system does not have repeated or mismatched knowledge. The knowledge extraction pipeline is illustrated in Figure 3.1.

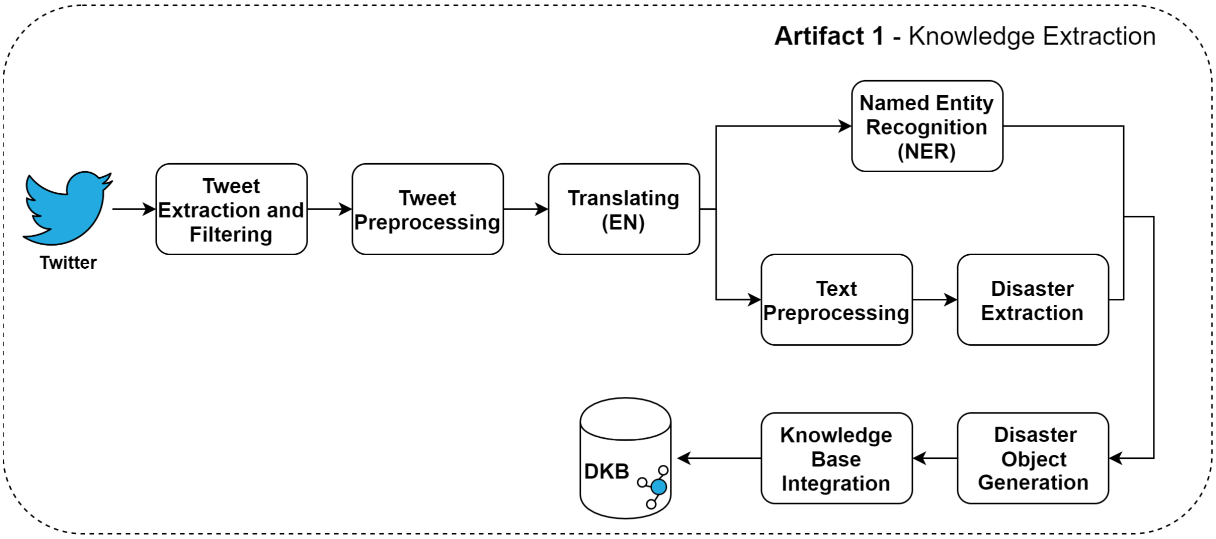
Artifact 1 - Knowledge Extraction
Twitter
Tweet Extraction and Filtering
Tweet Preprocessing
Translating (EN)
Named Entity Recognition (NER)
Text Preprocessing
Disaster Extraction
Disaster Object Generation
Knowledge Base Integration
DKB

FIGURE 3.1: Artifact 1 - Disaster extraction framework

3.2 Development

We will now discuss the implementation of this dissertation's first artifact. This subsection will describe in detail the created pipeline for knowledge extraction, as well as the data, tools and techniques that were used to create it. Each component is going to be separately depicted, nevertheless describing both inputs and outputs that connect that component with the remaining pipeline.

3.2.1 Tweet Extraction and Filtering

The extraction and filtering of our system's raw material, tweets, is of the utmost importance to guarantee that the system is not overloaded with meaningless data. To further minimize both performance and API limitations, this component makes periodic requests, 30 seconds apart, for new information. As for the request filters, only two are currently being used, the first being the unique identifier of the last gathered filter, to help prevent repeated information from being inserted on the system. The second, and most important filter is the geolocation of the area we want to gather tweets from. This information is created using Geopy[1], a Python client for several geocoding web services. We use this client to reverse search a geocode from a location name and a perimeter radius, using data from OpenStreetMap [38], a free world-map created using a collaborative peer production model.

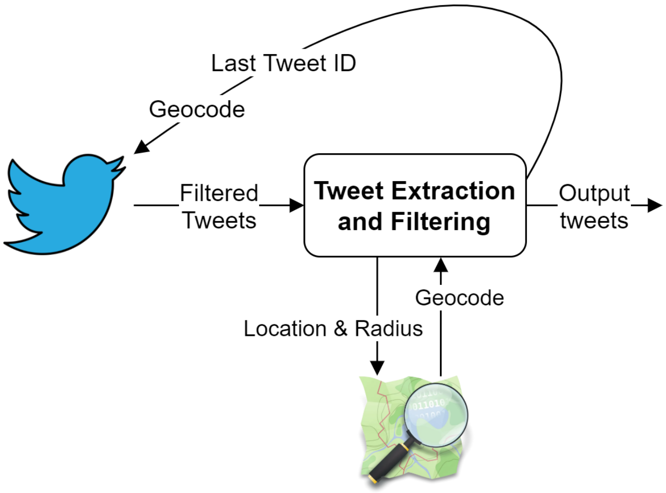

FIGURE 3.2: Tweet extraction scheme

3.2.2 Tweet Preprocessing

This module receives raw tweets as its input. This information is composed of both the textual content of a tweet and its metadata, such as the time it was posted at, its geolocation (if available), etc.

As we have previously mentioned, one of the challenges of using social media information is the vast amount of noisy data it pertains. Thus, tweet preprocessing refers to cleaning textual data that is very specific to tweets, that is, retweeting information, hashtags, hyperlinks, and user mentions. To approach this challenge, and seeing that this noisy information follows very specific patterns, the following approaches, based on hand-made regular expressions, were developed:

- **Retweet information** - Retweeting information, if present, always happens at the beginning of a tweet. It starts with the capital letters RT, followed by the nickname

---

[1] https://github.com/geopy/geopy

of the original user and ending with the colon symbol. To catch any occurrences of retweeting information, we use the regular expression:

```
RT(.*?):
```

This expression catches any range of text starting with the capitals RT and ending with a colon. As our system does not make use of such information, we apply the expression to find and delete it from the textual content of each tweet.

- **User mentions** - User mentions always start with an at symbol, followed by the user nickname, and can happen anywhere in the tweet's content. To catch these occurrences, we use the regular expression:

  ```
  @[^\s]+
  ```

  This expression matches the symbol `@` followed by any sequence of one or more characters that are not whitespaces. This ensures that our system catches and removes any occurrences of user mentions throughout the tweet's text.

- **Hyperlinks** - Hyperlinks can be expressed in many ways. They usually start with `www` or most commonly with one of the two protocols: `http` or `https`. The rest of the hyperlink follows, characteristically having symbols and letters, but no whitespaces. To match them, we use the regular expression:

  ```
  ((www\.[^\s]+)|(https?://[^\s]+)|(http?://[^\s]+))
  ```

  This expressions matches any sequence of symbols and letters starting with either `www`, `http://` or `https://`, selecting everything until the next whitespace it finds. The removal of hyperlinks is also essencial for our system, as this information has no value for the extraction of disasters and related information.

- **Hashtags** - Hashtags are a bit more relevant, as their content is generally defining of the whole tweet's content. To deal with them, we have used the two following regular expressions:

`(#[a-z1-9]+)` `([A-Z][a-z]+)`

> The first expression ensures our system correctly ignores lower-case hashtags, as these often have several interconnected lower-case words and can prove to be troublesome for the system. The second expression matches all words starting with a capital letter followed by a set of lower-case letters. This is used to split all words connected with capital letters, thus splitting hashtags in the format #FirstSecond into # First Second. After this process, all # occurrences in the tweet are removed.

After these transformations have been made, the tweet's text is sent to the next module of the system. Figure 3.3 displays a practical example of this component's application.

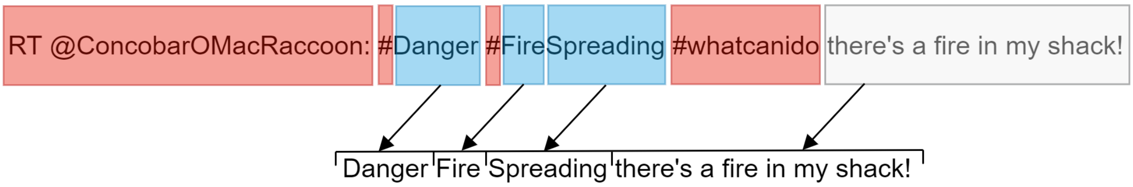

FIGURE 3.3: Tweet preprocessing example

3.2.3 Translating

Translation is a natural step in our system's pipeline, as it grants it more flexibility in terms of who can use it, and where it can be implemented. Besides this, most state-of-the-art models for data extraction are trained in English, or depend on vectorial representations of English words. In view of this, a translating component has been integrated in our system. To give it further flexibility, the language of the input text is automatically detected, and then its content is translated to English. This is done using Google's Neural Machine Translation (GNMT) [39], which consists in a deep Long Short-Term Memory (LSTM) Neural Network, using both residual and attention connections. This model has achieved a BLEU [40] score of 38.95 in the commonly evaluated en-fr translation task, with a vocabulary size of around 32,000 words. It is also reported to reduce translation errors by more than 60%, when compared to Phrase-based Machine Translation (PBMT) models [39].

Although translating tweets can, in part, have a negative impact in their coherence, most translation issues are related to the sentence construction, which is primarily done using common stop words. As stop words are also commonly removed as a preprocessing step for most NLP extraction and classification algorithms, and we translate sentences after removing twitter-related noise, we believe that the impact of translating tweets will be minimum, and worth the benefit of the added flexibility it offers the system.

3.2.4 Named Entity Recognition

NER is used as a tool to label and extract entities from the aforementioned tweets. These entities, in general, can represent person names, companies, geographical locations, times, and dates. Their role is to complement a detected disaster, in order to give the population and first-responders more insight. For this purpose, our system makes use of SpaCy's NER pipeline. SpaCy's main English NER model is a multilayer Convolutional Neural Network (CNN) trained using data from OntoNotes[2], vectorized using GloVe embeddings[3].

Although it provides one of the best NER models out there, using SpaCy also brings some issues. SpaCy's NER training data is mostly referring to the United States of America (USA), so the models are very well suited for detecting American locations, but only some more generic locations outside of its borders. For our scenario however, where we are trying to create a system that is able to help local population in disaster scenarios, local knowledge, such as organizations and locations, is of great importance.

To overcome this stumbling block, we have implemented a new layer on top of SpaCy's NER pipeline. This layer is responsible for labeling and extracting any local knowledge that we manually model, which can either be SpaCy's standard NER entities like PERSON, or even custom entities we may want to implement.

---

[2] https://catalog.ldc.upenn.edu/LDC2013T19

[3] https://nlp.stanford.edu/projects/glove/

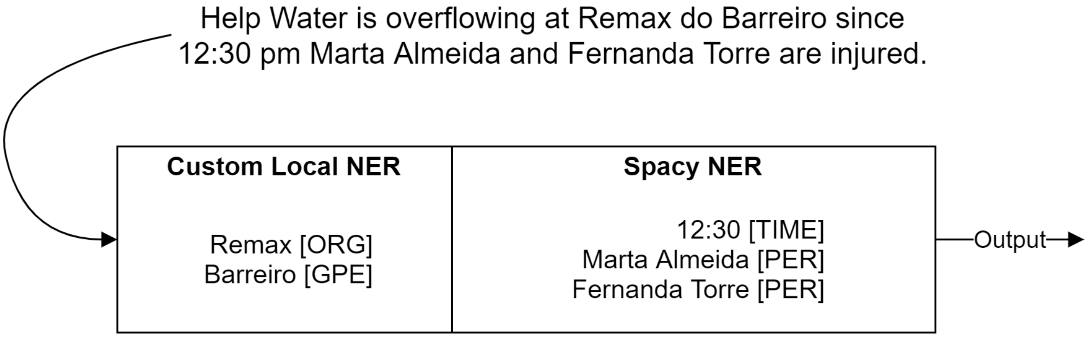
Help Water is overflowing at Remax do Barreiro since 12:30 pm Marta Almeida and Fernanda Torre are injured.
Custom Local NER
Remax [ORG]
Barreiro [GPE]
Spacy NER
12:30 [TIME]
Marta Almeida [PER]
Fernanda Torre [PER]
Output

FIGURE 3.4: Named Entity Recognition using Local and SpaCy Knowledge

3.2.5 Text Preprocessing

Preprocessing is an essential step for natural language algorithms. Although activities like Named Entity Recognition (NER) benefit from inputting entire texts with punctuation and the original formatting, most algorithms benefit from further processed text. This step focuses on additionally preparing tweets for Disaster Extraction.

Our simple preprocessing pipeline starts by removing any punctuation from the tweets, as it does not contribute for the classification of disasters. This is done using a simple regular expression that matches anything that is neither a word nor a whitespace:

```
[^\w\s]
```

Tokenizing the tweets is the next natural step, and it is done using Natural Language Toolkit[4](NLTK) algorithms. Next, all words are lowercased, to guarantee that they are normalized for the next steps in our pipeline. After we have split and lowercased all words, stop-words – the most common words in a language – are detected and removed. These words are so common in texts that, for most tasks, very rarely offer any added value to the classification models, frequently being regarded as noise that can lead to a worse performance. This is done by comparing all of the tweet words with NLTK's list of 178 english stop-words, and then removing any matches.

---

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

Even though lemmatization and stemming approaches are also very common, we have decided not to use them. Our predictive models have shown a drop of 1 to 5 p.p in their results, depending on the model and measure of estimation, when using any of these techniques. Consequently, our disaster extraction predictive models have slightly benefited from not using them.

3.2.6 Disaster Extraction

Disaster extraction is, without any doubt, one of the most critical steps of our system, as all of its knowledge and functionalities are centered in these disasters. It consists in two sequential tasks: Firstly, classifying tweet relatedness to disasters, that is, classifying if a tweet's content is referring to a disaster event. Secondly, classifying the disaster type a tweet is referring to. To accomplish both of these tasks we have implemented two distinct Convolutional Neural Networks (CNNs), as it has proven to provide better classification results, when compared to only using a CNN for the second task. The first CNN takes care of the first task, and if it classifies the tweet as being related to disasters, the second CNN then classifies which disaster it refers to, out of a predefined set of disasters it has been trained with, which are fire, flood, earthquake, explosion and none. These classes have been selected due to their relevance and frequency in real-world scenarios, as well as the existence of enough previously classified data related to them. The last category exists as a double-check, given that the first CNN can detect disaster events that are not modelled by the second CNN. The second neural network outputs a list of confidence levels regarding all disaster classes.

In the event of the second CNN classifying a tweet with close confidence levels, that is, two types of disasters being very closely related with the content of the tweet, a weighted check is triggered. When we refer to having close confidence levels we mean $\text{ConfDisaster1} - \text{ConfDisaster2} < \text{ConfMargin}$, where Conf1 refers to the highest confidence the CNN outputs, Conf2 refers to the second highest confidence, and ConfMargin refers to the minimum margin of confidence the model should have betweeen the top two disasters. This check consists in adding to each of this confidence levels the confidence of a lexicon-based model.

The lexicon-based model is a very simple algorithm which consists in – after pre-processing the original tweet with tokenization and stemming techniques – comparing the words of the tweet with a predefined lexicon for each type of disaster. The results, LexCount, then go through a min-max normalization:

$$\text{LexCount'} = \frac{\text{LexCount- min(LexCount)}}{\text{max(LexCount)- min(LexCount)}} \tag{3.1}$$

Lastly, a weighted ponderation of these values is added to the output confidence of the neural network, allowing for a better classification of the disaster a tweet refers to. This component's pipeline is also represented in Figure 3.5.

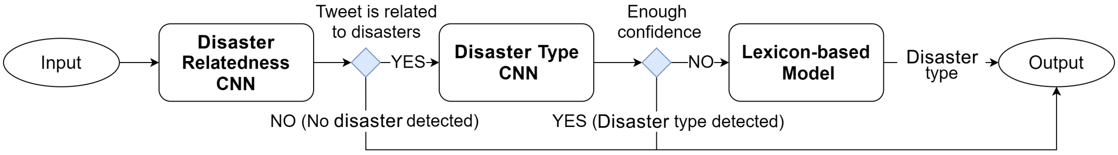

Figure 3.5: Disaster extraction component pipeline

The data used to train both networks was obtained as a subset of two CrisisLex [41] disaster datasets:

1. CrisisLexT6[5] - Approximately 250,000 tweets posted during 26 crisis events in 2012 and 2013, labeled by informativeness, information type and source;
2. CrisisLexT26[6] - Approximately 60,000 tweets posted during 6 crisis events in 2012 and 2013, labeled by relatedness (as "on-topic", or "off-topic") with each event.

Although they do not have the same data, as a set, they have English tweets across more than 30 large disasters which took place between 2012 and 2013. These include, for

---

[5] https://crisislex.org/data-collections.htmlCrisisLexT6

[6] https://crisislex.org/data-collections.htmlCrisisLexT26

example, the 2012 Colorado Wildfires and the famous 2013 Boston Marathon Bombings. On top of this, to better represent the semantics of the domain where they are being applied, our classification models use the 200d Glove [42] Twitter embeddings[7]. These pretrained embeddings have explicitly been trained from more than 2 billion tweets, with about 30 billion tokens, and more than 1.2 million unique vocabulary, where each word is represented as a 200-dimensional vector. The usage of these embeddings is a huge advantage in tweet information extraction, as most pretrained embeddings are either trained with Wikipedia content, web crawling data, or news.

CNN Architecture and Test Results As shown in Figure 3.6, our CNNs have an embedding layer, followed by the convolutional layer with a Rectified Linear Unit (ReLU) activation and five similar filters, followed by a global max pooling layer. The network then has a set of three dense layers – that also use ReLU activation – with decreasing number of nodes and dropout rates, which help us prevent overfitting our data. This is extremely important because we are dealing with data relating to particular scenarios/disasters, so we want to keep our models from overfitting specific data, which can be done, to some extent, by ignoring random neurons during the training phase. The difference between our two models is in the last layer. The output is yielded through a dense layer, which varies from having one node and a sigmoid activation in the relatedness classification CNN, and having five nodes and a softmax activation in the type classification CNN.

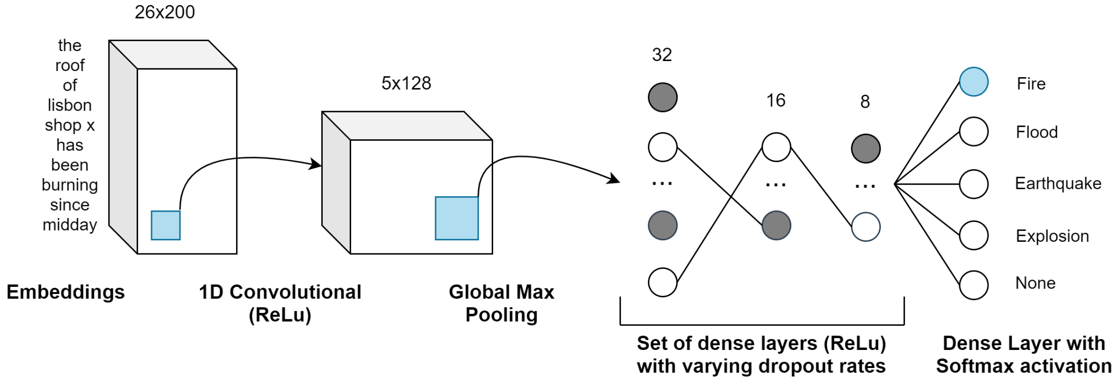
26x200
the roof of lisbon shop x has been burning since midday
5x128
32
16
8
...
Fire
Flood
Earthquake
Explosion
None
Embeddings
1D Convolutional (ReLu)
Global Max Pooling
Set of dense layers (ReLu) with varying dropout rates
Dense Layer with Softmax activation

FIGURE 3.6: Disaster type classification CNN

---

[7] https://nlp.stanford.edu/projects/glove/

To deal with the overfitting issue of the models, besides inserting dropouts and tweaking some of the hyperparameters, we have used an Early Stopping strategy. As we can see in Figures 3.7 and 3.8, the models tend to overfit the training data, thus showing an increase in the validation loss of the models, and continuing decrease in the training loss. To prevent this, the Early Stopping method consists in monitoring this process and stop training once the model performance on the validation dataset stops improving. Taking this into account, our first CNN was trained along three epochs, while the second was trained along nine epochs.

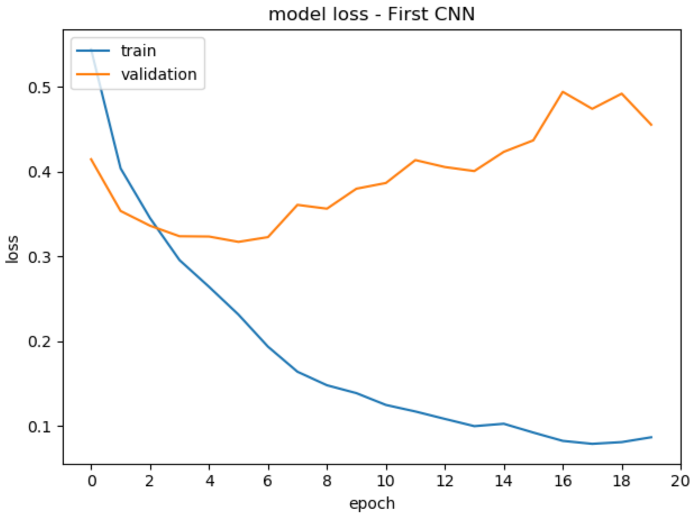

Figure 3.7: Loss evolution through the training epochs of the first CNN (Related Classification)

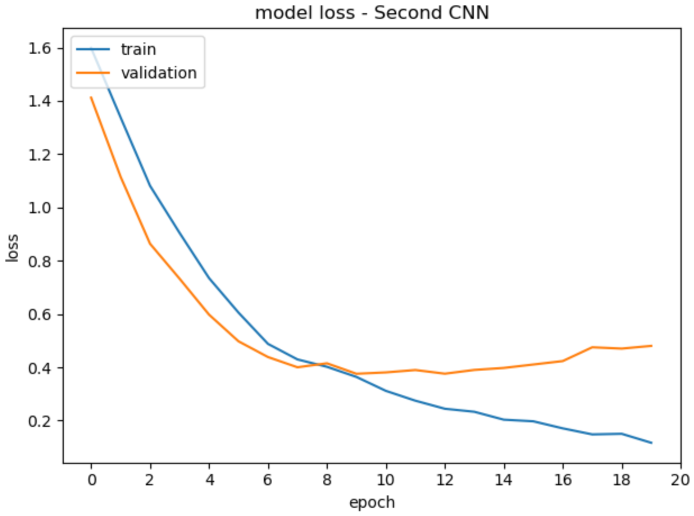

Figure 3.8: Loss evolution through the training epochs of the second CNN (Type Classification)

For the disaster type classification CNN, we have also tried an alternative architecture. Although published in 2014, Kim [43] CNN for sentence classification has thousands of citations and is still, in 2020, cited in hundreds of works. This architecture showed that CNNs, which were until then mostly used for image classification, can display state-of-the-art results for sentence classification. Figure 3.9 displays one of Yoon Kim's proposed CNN architectures.

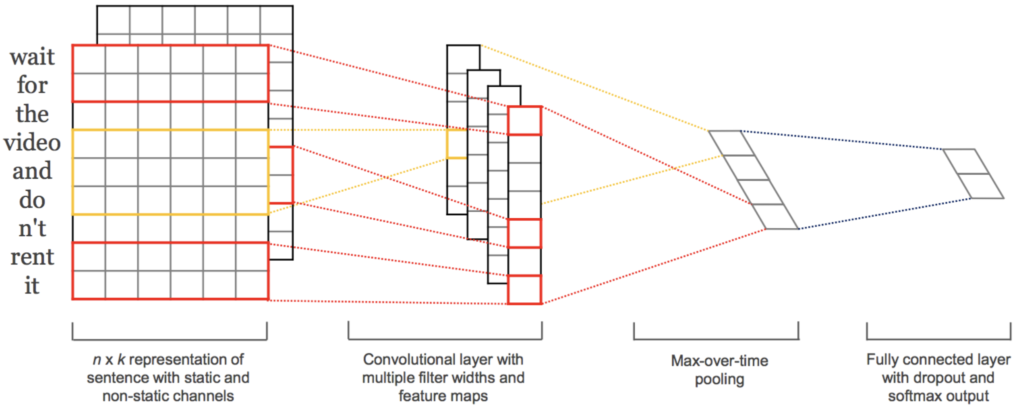
wait
for
the
video
and
do
n't
rent
it
n x *k* representation of sentence with static and non-static channels
Convolutional layer with multiple filter widths and feature maps
Max-over-time pooling
Fully connected layer with dropout and softmax output

FIGURE 3.9: Yoon Kim model architecture with two channels, Kim [43]

To compare our models with the current state-of-the-art literature on tweet disaster classification, we have divided this information in the two aforementioned tasks: event relatedness classification and event type classification. We have then selected the best results of each relevant publication, being essential to highlight that all of these have been published between 2018 and 2019. The compared models use different algorithms and implementation techniques, but all of them use subsets of the same datasets – crisisLex – to train.

Although not all the compared studies have shown their Accuracy, Precision or Recall, we have decided to also include these measures, as they may prove to be interesting in comparison with future works. To compare with the state-of-the-art however, as it is the most common measure for text classification, we use the F1-Score (Eq. 3.2). This measure considers both precision and recall, and is often preferred in the field of information retrieval, as it offers a better estimate of wrongly classified cases, when compared to accuracy.

$$\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3.2}$$

Our models have been trained using a subset of close to 20,000 labeled tweets. These tweets were balanced for the relatedness classification task, having around 54% related tweets and the remaining 46% unrelated. For the second task, however, there was a need to use under-sampling techniques to balance major classes – as our models were greatly overfitting those same classes – ending up with close to 5,000 tweets. Besides this, there were also some preprocessing steps involved before feeding the tweets to the embedding layer of the networks. The preprocessing consisted in the lowercasing, tokenization and stop-word removal of the tweets, using the same techniques that were discussed in subsection 3.2.5. In addition all tweets were padded, to normalize all of the inputs used to train the models. As for the split of the data, we went with a fixed ratio of 75, 10 ,15 for training, validation and test respectively.

The first CNN related classifier has fared exceptionally well in relatedness classification task. We have achieved an increase of almost 10 p.p of F1-Score when compared to the current state-of-the-art [32], with a test F1-Score of 92. This improvement can be explained by the usage of Twitter-specific embeddings, preprocessing methods, and variations in the model's architecture. Detailed results are shown in Table 3.1.

Model	**Accuracy**	**Precision**	**Recall**	**F1-Score**
CNN Burel et al. [32]	-	-	-	0.838
LSTM Sit et al. [33]	0.748	-	-	0.751
CNN	0.903	0.921	91.9	0.920

Table 3.1: Comparative results of the event relatedness classification task

As for the second task, disaster type classification, our simple CNN architecture has achieved an improvement of 1 to 10 pp when compared to our implementation of Kim [43] static architecture, using the same word embeddings and preprocessing techniques. When compared to the state-of-the-art, although not as good, our CNN has achieved good results. An investigation on these results led us to believe that our type-classification CNN has a slight overfit to the Flood class, sometimes misclassifying unrelated tweets. This problem is, however, fixed with the lexicon-based model that was previously presented, although not reflected in these results, as to correctly compare the models. Detailed results are shown in Table 3.2.

Model	**Accuracy**	**Precision**	**Recall**	**F1-Score**
SVM Burel et al. [32]	-	-	-	0.997
CNN Alam et al. [34]	0.930	-	-	0.928
CNN	0.884	0.888	0.876	0.882
CNN Kim [43]	0.847	0.879	0.776	0.823

Table 3.2: Comparative results of the event type classification task

3.2.7 Disaster Object Generation

The second to last step of our pipeline consists of joining all of the information extracted from the previous steps. That is, the tweet information extracted by the Tweet Extraction and Filtering module, all of the entities extracted by the Named Entity Recognition module, and the disaster type classified by the Disaster Extraction, are all combined to form a disaster object. This step is essential, as it creates a sense of identity for the detected disasters, joining their occurrences to all of the information that defines them. We consider the location, date and type of disaster to be the minimum necessary information for creating and comparing a disaster. Therefore, this module also serves as an extra filtering phase, as it discards any disasters the system has not been able to extract enough information from.

3.2.8 Knowledge Base Integration

To understand the knowledge integration, we must first understand how the knowledge base itself works.

As our system requires a way to efficiently store and connect data, a knowledge base is needed. This storage can be addressed by database technologies, such as graphs. Graphs are extremely capable when extracting insight from data is a top priority, as they are mainly made of nodes and connections, and translating that to the real world, it means we can have access to a logical map of entities and their relationships. Their structure also allows for high flexibility when expanding to incorporate new data. When thinking about the development and future improvements of the system, graphs are also a great choice, as they align perfectly with today's agile, test-driven development practices, allowing our knowledge base to evolve with the rest of the application and any changing requirements. Besides their structure, graphs are also very efficient performance-wise, as in contrast to traditional databases, their performance stays nearly constant even as their data grows.

Bearing in mind the stated advantages, our knowledge base consists of a Dynamic Knowledge Graph [44]. This means we can structure entities as a knowledge graph in which new nodes and relationships are constantly being added as new context is detected, to expand and upgrade its knowledge in real-time. This knowledge is disaster-centered, as disaster occurrences are represented by specific nodes which are then connect to locations, dates, people, organizations etc. Besides having several node types, such as the disasters and locations, each node also has several attributes. This allows for flexibility when dealing with various types of data associated with entities. As for its management, our knowledge base uses the Neo4j[8] graph database management system. The interactions with it are done through in-house developed custom methods, using a mixture of the Cypher query language, and the Py2neo[9] toolkit. As an example of how Cypher querying works, given a disaster type T, a disaster location L and a disaster date D, a Cypher query to check if a disaster node already exists would be:

```
MATCH (n:Disaster)
MATCH (n)-[:IS]->(t:Type) WHERE t.name = "T"
MATCH (n)-[:LOCATED_IN]->(l:Location) WHERE l.name = "L"
MATCH (n)-[:STARTED_ON]->(d:Date) WHERE d.name = "D"
RETURN COUNT (n)
```

---

[8] https://neo4j.com/

[9] https://py2neo.org/v4/

The pipeline step of integrating new knowledge in the knowledge involves all necessary validations. They can be regarding to the previous existence of our gathered data, as well as the creation of new nodes and relationships between them, to express the identity of the extracted disasters in a way that facilitates human consumption and future system integrations. The chart flow represented in Figure 3.10 represents the flow of validations and resulting actions when trying to register a new disaster object in the Knowledge Base.

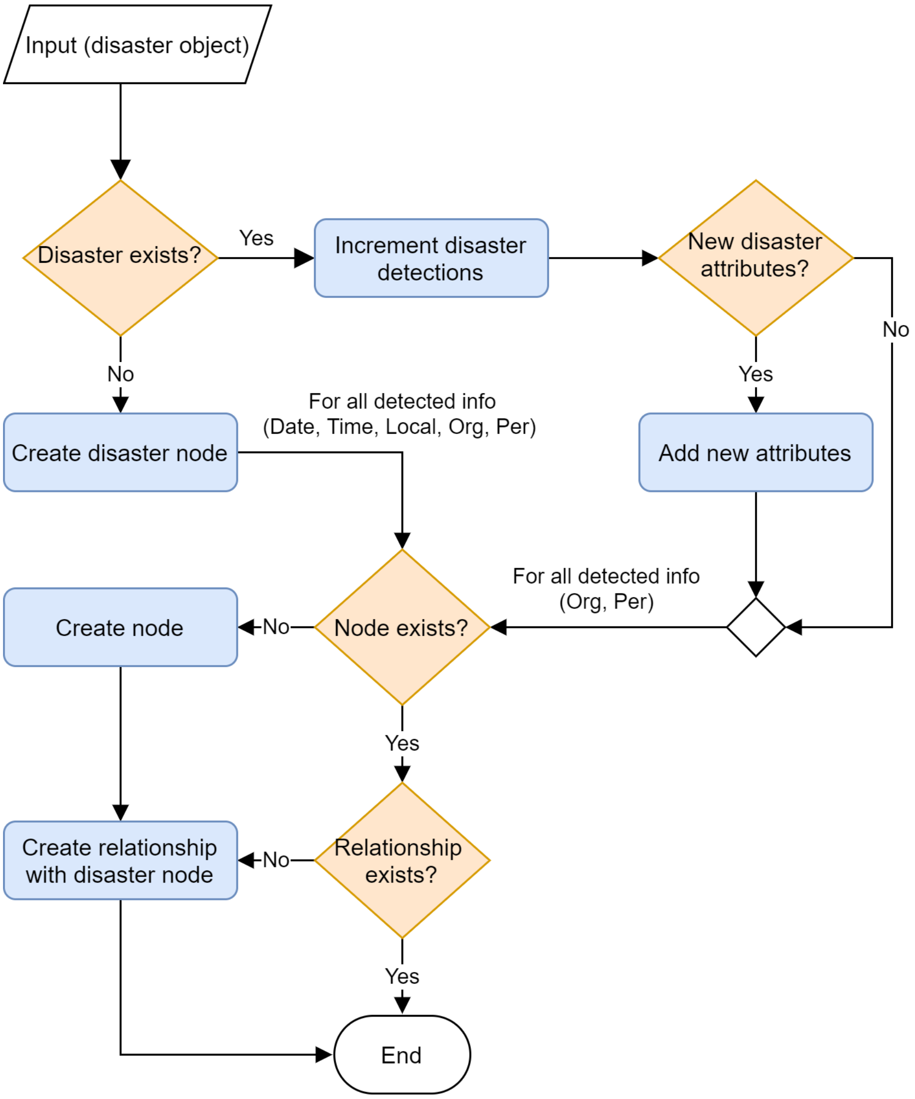

FIGURE 3.10: Simplified Knowledge Base integration flow chart

The most important validation that is done, when a disaster is detected, is to check if it already exists in the knowledge base. This is done by checking for the existence of a disaster node with relationships to nodes representing the extracted date, location and disaster type. Only if all of these conditions are verified does the system acknowledge the presence of the extracted disaster object in the knowledge base.

To further deal with social media unreliability, all disaster nodes have an attribute which is incremented when the same disaster is detected from different tweets. This attribute can then be used as a detection threshold, to allow for better filtering of relevant disasters. If new information regarding an already existing disaster is extracted from the previous steps, it is also validated and then connected with that disaster. This procedure allows for our system to be continually learning, expanding its knowledge, and bettering the knowledge it already has.

Chapter 4

Artifact 2 – Design & Development

4.1 Design

The second artifact consists in how the system communicates the extracted information to its end users. The Digital Assistant, otherwise known as chatbot, receives messages and sends replies accordingly. In between these happenings, this artifact 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[1].

After the messages have been preprocessed, our 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. Afterwards, 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, as well as 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

---

[1] https://spacy.io/

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 4.1.

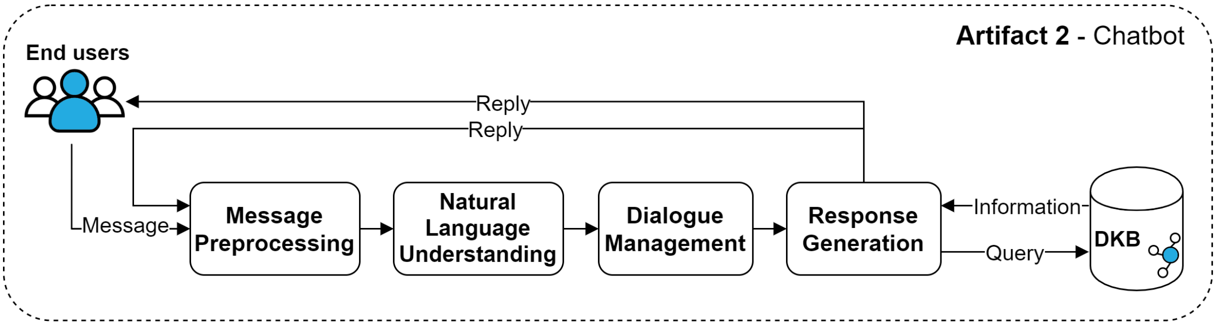

FIGURE 4.1: Artifact 2 - Chatbot framework

4.2 Development

One of the main tools used in building our second artifact, the disaster-support chatbot, was RASA[2]. RASA is a machine learning framework for text- and voice-based assistants automation that allows for a higher level of abstraction when creating chatbots. We have specifically used RASA Open Source, the solution's free version which, although a bit 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, like Google's Dialogflow[3] or Facebook's Wit.ai[4], 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[5] featurizers and Facebook's Duckling[6], which is a Haskell library that parses text into structured data.

---

[2] https://rasa.com/

[3] https://cloud.google.com/dialogflow

[4] https://wit.ai/

[5] https://scikit-learn.org/stable/

[6] https://github.com/facebook/duckling

4.2.1 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 step in this preprocessing pipeline 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 for certain NLP tasks, all text is valuable, we have not used other common preprocessing methods, such as stop-word removal, lemmatization etc. The second and last step of this pipeline 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 4.1 specifies all of these featurizers and their usage.

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

TABLE 4.1: Used featurizers and related functions

4.2.2 Natural Language Understanding

The NLU component of our chatbot 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 his/her message's intention, 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 questions correctly, as it would have no means to understand whose contact the user needed.

Both intents and entities are being classified using the state-of-the-art architecture Dual Intent Entity Transformer (DIET) [45]. The DIET classifier has been 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 training time and compute-intensive.

The DIET architecture is based on a Transformer [46] shared for both tasks. A sequence of entity labels is predicted by a Conditional Random Field (CRF) [47] tagging layer, on top of the transformer output sequence corresponding to the input sequence of tokens. The intent labels the transformer outputs for the classification token `_CLS_` are embedded into a single semantic vector space. A dot-product loss is then used to maximize the similarity with the target label and minimize similarities with negative samples. The schematic representation of DIET's architecture is presented in Figure 4.2.

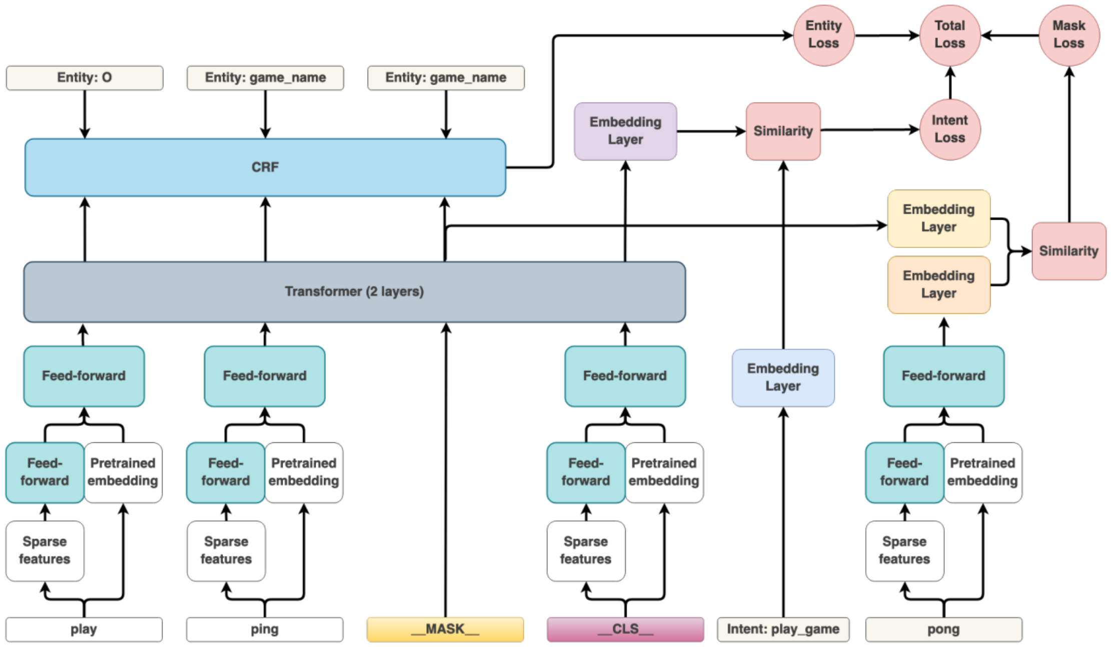

FIGURE 4.2: Schematic representation of the DIET architecture, Bunk et al. [45]

Our NLU component has been trained using more than 400 hand-made annotated phrases that we have created, divided into 25 distinct intents. These phrases also include more than 100 entity references for 4 distinct custom entities: time, location, organization, and disaster type. Our intents are divided into generic intents, chitchat intents, and request intents. Table 4.2 presents all of these intents, with some examples:

Category	**Intent**	**Unannotated Example**
Generic	greet	Olá
	goodbye	até breve
	thanks	obrigado pela ajuda
	affirm	sim
	deny	nem pensar
	stop	já chega
	inform	há 5 dias atrás
Chitchat	chitchat_how_are_you	tudo bem?
	chitchat_question	tens a certeza?
	chitchat_bot_challenge	eu sei que não és humano
	chitchat_insult	és inútil
	chitchat_weather	como está o tempo hoje?
	chitchat_ask_name	como é que te chamas?
	chitchat_where_are_you	moras em Portugal?
	chitchat_nutrition	gostas de comida vegan?
	chitchat_animal	tens animais de estimação?
	chitchat_random_questions_yes	gostas do teu trabalho?
	chitchat_random_questions_no	queres casar comigo?
Help	help_general	socorro!
	help_contact	preciso de ajuda, como posso falar com os bombeiros do Barreiro?
	disaster_query_active	há algum incêndio em Lisboa?
	disaster_query_past	há informação sobre cheias este mês na zona do Barreiro?
	disaster_todo	devo sair de casa durante um terramoto?
	disaster_inform	há um incêndio onde eu vivo!
	organization_state	posso utilizar o centro de saúde do lavradio?

TABLE 4.2: Chatbot implemented intents with unannotated examples

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.

4.2.3 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:

Slots and Forms Slots represent part of our chatbots memory. They act as key-value storage that can save both user-provided information, like a location's name, or chatbot-inferred information, like knowledge base query results. Although this is not always true, slots tend to influence the dialogue progression. One such example is forms 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.

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. Our chatbot has a few dozens of stories which try to simulate several possible conversations inside and outside of a disaster scenario, in order to train our chatbot for both expected and unexpected scenarios. One simple example of a story, where the user asks for an organization's contact, is the following:

```
* greet
  - action_daytime_greet
* help_contact{"organization": "Proteção Civil"}
  - action_get_contact
```

```
* thanks
  - utter_no_problem
* goodbye
  - utter_goodbye
```

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, filled slots etc. Usually chatbots have several policies combined. Our chatbot 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 5-turns 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 our chatbot 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 the others, it does not have a niche scenario where it is applied. The Transformer Embedding Dialogue (TED) policy [48] 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 5 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 [49].

4.2.4 Response Generation

Dialogue Management is responsible for selecting the chatbot'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 our chatbot runs in response to an user's intent, and in our chatbot, three types of actions are being used:

Utterance Actions These are the simplest types of actions. What they do, is 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 the 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 refering to?
- Está-se a referir a quando?/What time are you refering to?
- Em que altura?/When?

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.

Custom Actions Custom actions are where our Graph Knowledge Base, and other external knowledge sources, come in. 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. Our chatbot currently has nine distinct custom actions, which serve different purposes. They can go from very simple actions, like 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, like 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.

As mentioned before, the way the chatbot asks the user to confirm the intent, is originally done using a default action. We have, however, overridden this action with a custom action of our own, because we want these confirmation requests to change depending on the intent itself. Figure 4.3 shows the process of building the first reply of the Two-stage Fallback Policy's overridden default action.

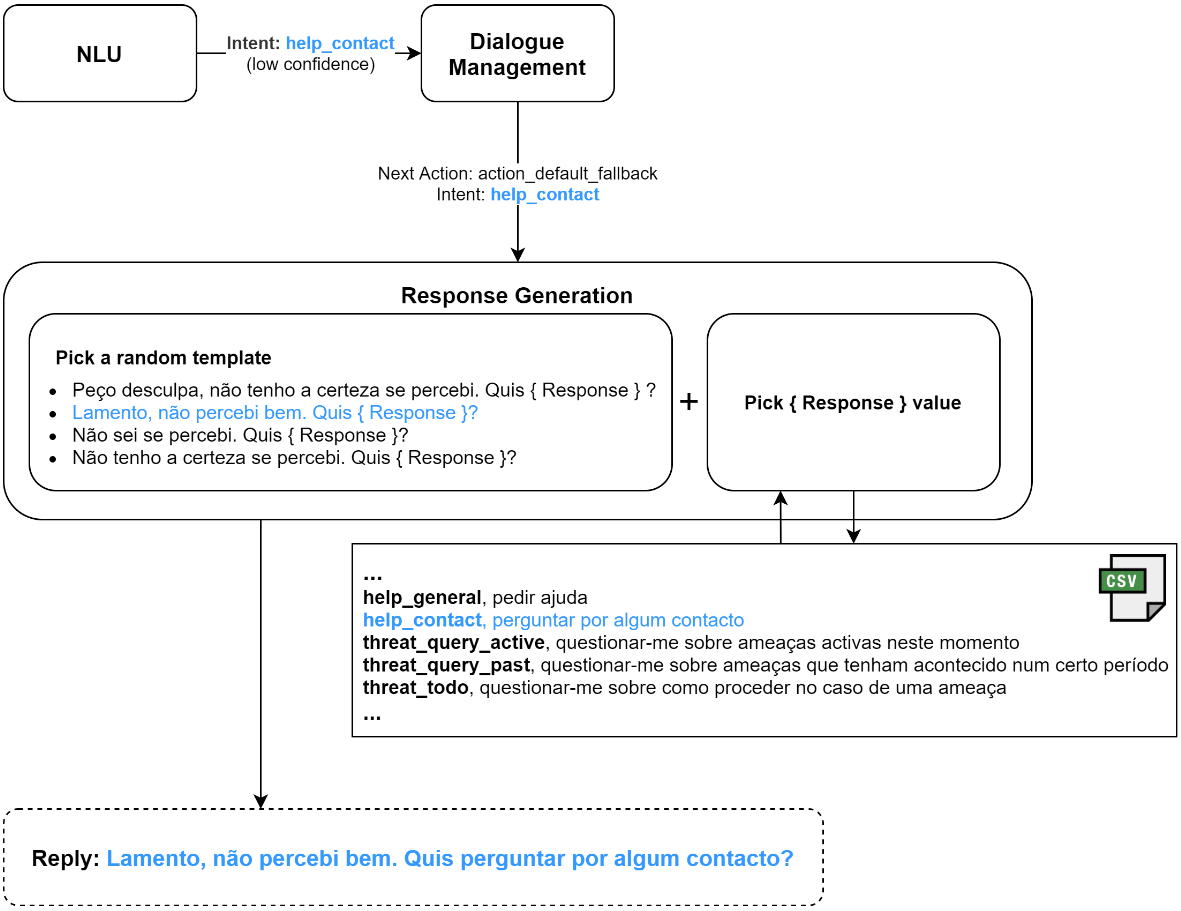
NLU
Intent: help_contact
(low confidence)
Dialogue Management
Next Action: action_default_fallback
Intent: help_contact
Response Generation
Pick a random template
Peço desculpa, não tenho a certeza se percebi. Quis { Response } ?
Lamento, não percebi bem. Quis { Response }?
Não sei se percebi. Quis { Response }?
Não tenho a certeza se percebi. Quis { Response }?
+
Pick { Response } value
CSV
...
help_general, pedir ajuda
help_contact, perguntar por algum contacto
threat_query_active, questionar-me sobre ameaças activas neste momento
threat_query_past, questionar-me sobre ameaças que tenham acontecido num certo período
threat_todo, questionar-me sobre como proceder no caso de uma ameaça
...
Reply: Lamento, não percebi bem. Quis perguntar por algum contacto?

FIGURE 4.3: Building the first reply of a Fallback action

Some other examples of this type of actions, are the actions that infer the DKB for disaster-related knowledge. This knowledge has been created in our first artifact, and set in the DKB, which is the bridge between both artifacts. As mentioned before, our chatbot uses that knowledge to be able to inform citizens 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, much like the `help_contact` action in Figure 4.3. Imagining the user provides all required information in his interaction and makes the request on the 22nd of August, Figure 4.4 portrays how the chatbot would interact with the DKB to build its reply.

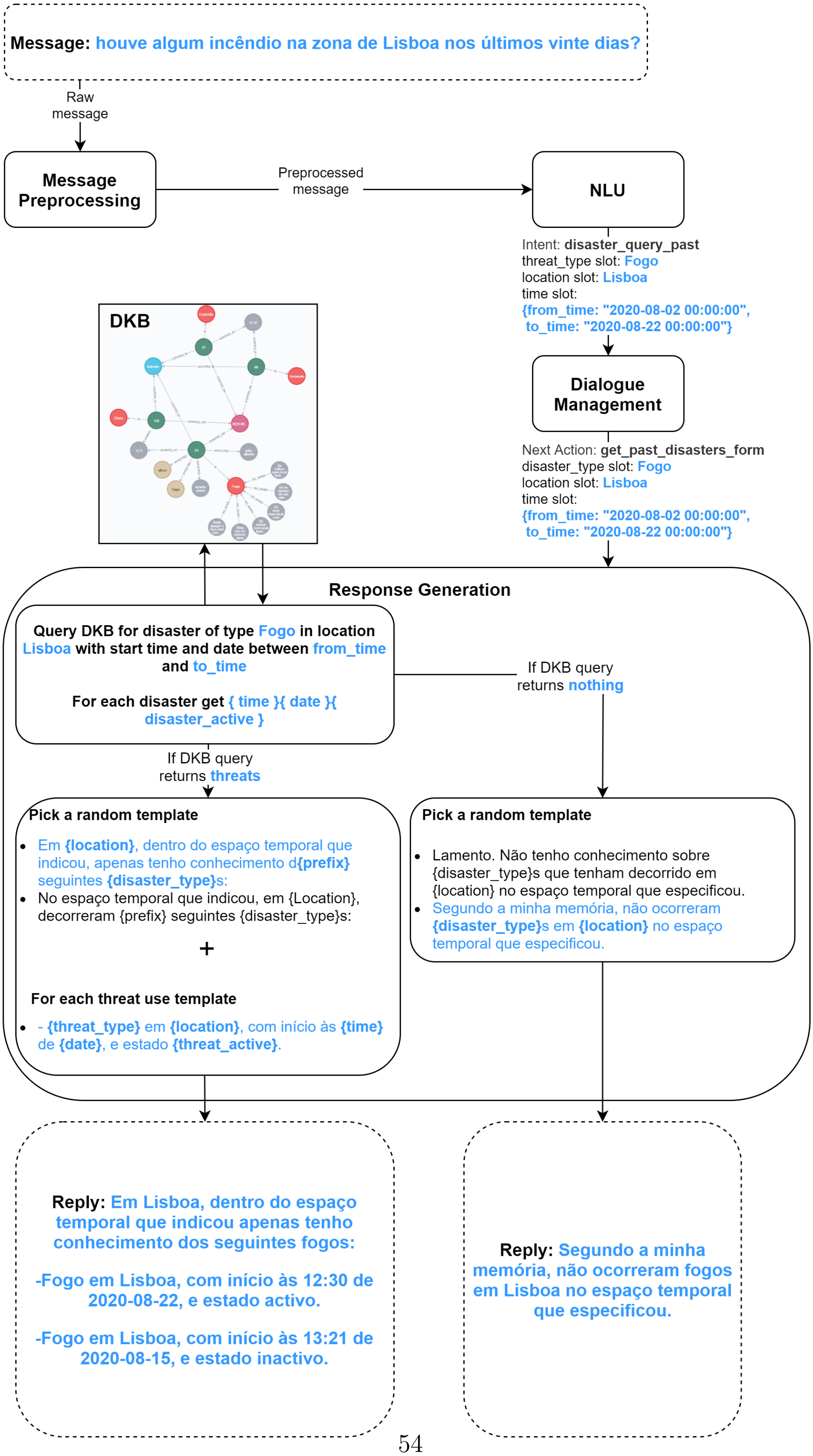
Message: houve algum incêndio na zona de Lisboa nos últimos vinte dias?
Raw message
Message Preprocessing
Preprocessed message
NLU
Intent: disaster_query_past
threat_type slot: Fogo
location slot: Lisboa
time slot:
{from_time: "2020-08-02 00:00:00",
to_time: "2020-08-22 00:00:00"}
Dialogue Management
Next Action: get_past_disasters_form
disaster_type slot: Fogo
location slot: Lisboa
time slot:
{from_time: "2020-08-02 00:00:00",
to_time: "2020-08-22 00:00:00"}
DKB
Response Generation
Query DKB for disaster of type Fogo in location Lisboa with start time and date between from_time and to_time
For each disaster get { time }{ date }{ disaster_active }
If DKB query returns nothing
If DKB query returns threats
Pick a random template
Em {location}, dentro do espaço temporal que indicou, apenas tenho conhecimento d{prefix} seguintes {disaster_type}s:
No espaço temporal que indicou, em {Location}, decorreram {prefix} seguintes {disaster_type}s:
+
For each threat use template
- {threat_type} em {location}, com início às {time} de {date}, e estado {threat_active}.
Pick a random template
Lamento. Não tenho conhecimento sobre {disaster_type}s que tenham decorrido em {location} no espaço temporal que especificou.
Segundo a minha memória, não ocorreram {disaster_type}s em {location} no espaço temporal que especificou.
Reply: Em Lisboa, dentro do espaço temporal que indicou apenas tenho conhecimento dos seguintes fogos:
-Fogo em Lisboa, com início às 12:30 de 2020-08-22, e estado activo.
-Fogo em Lisboa, com início às 13:21 de 2020-08-15, e estado inactivo.
Reply: Segundo a minha memória, não ocorreram fogos em Lisboa no espaço temporal que especificou.

FIGURE 4.4: Chatbot response generation for the `get_past_disasters_form` action

Chapter 5

Demonstration & Evaluation

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

5.1 Demonstration Scenario

To demonstrate our evaluators – the three civil protection specialists, which represent possible end-users – what our artifacts do, and how well they do 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. This earthquake's cascading effects lead to an explosion in a tank filled with an inflammable substance, located in one of the municipality's SIPS, FISIPE. The resulting fire then spreads to the surrounding area, also reaching Alkion, another nearby SIPS, and 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 in the knowledge extraction system, as they are posted by the municipality's citizens that felt the earthquake, and citizens walking by in Barreiro's 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 50 tweets related to the described scenario, and created by local citizens. This request had the following guidelines:

- 25 generic and varied tweets reporting an earthquake in Barreiro;
- 15 generic and varied tweets about a fire in Barreiro;
- 5 tweets referring a fire happening in Barreiro's FISIPE;
- 3 tweets about the fire spreading to Barreiro's Alkion;
- 2 tweets about the fire having injured specific citizens.

A critical analysis of these tweets, has led us to classify them into two classes: related but not informative and related & informative. The first one refers to tweets that are related to the disaster scenario, but do not provide enough information to be useful for disaster management, usually by not specifying a location or type of disaster. One example of these tweets is "Não se consegue circular em muitas estradas do concelho/It is impossible to travel in the county's roads". On the other hand, the second class refers to related tweets that are informative and offer added value to the disaster management task, one example being "#EstáTremido As minhas paredes começaram a tremer… é um terramoto no Barreiro?/#ItsShaking My walls have begun to shake... is it an earthquake in Barreiro?". Out of the 50 provided tweets, 27 have been manually classified as being related & informative, and the remaining 23 as being related but not informative.

Besides these tweets, we have also gathered 750 local tweets – from within Barreiro and nearby areas – directly from Twitter, to serve as the always present social media noise. With the totality of 800 tweets, 6.25% forged and related to disasters, and the remaining 93.75% real unrelated tweets, we aim at representing, as close as possible, a real scenario. Appendices A and B display subsets of the related and unrelated tweets, respectively.

Figure 5.1 presents both the user-artifact interaction and the timeline for the demonstration use-case. A possible conversation with the chatbot, in accordance with this use-case, can also be seen in Appendix C.

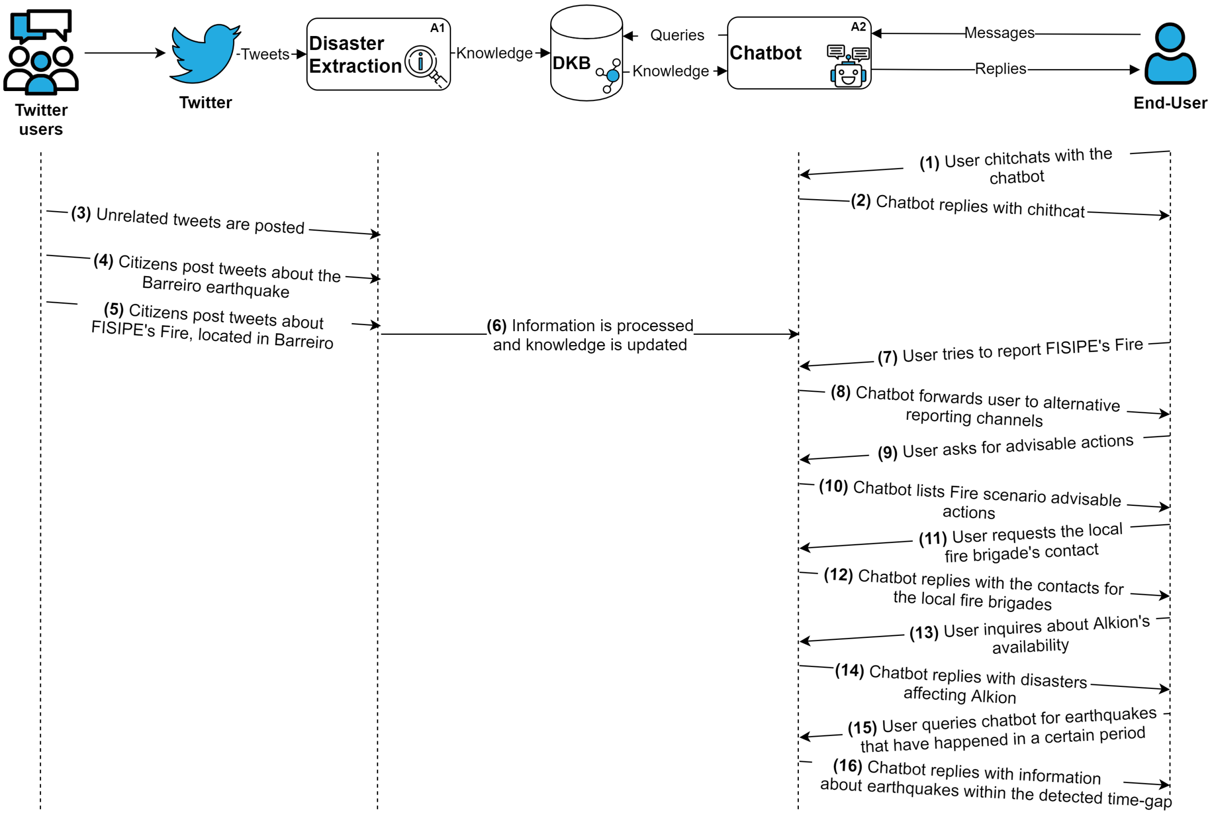
Twitter users
Twitter
Tweets
Disaster Extraction
A1
Knowledge
DKB
Queries
Knowledge
Chatbot
A2
Messages
Replies
End-User
(1) User chitchats with the chatbot
(2) Chatbot replies with chithcat
(3) Unrelated tweets are posted
(4) Citizens post tweets about the Barreiro earthquake
(5) Citizens post tweets about FISIPE's Fire, located in Barreiro
(6) Information is processed and knowledge is updated
(7) User tries to report FISIPE's Fire
(8) Chatbot forwards user to alternative reporting channels
(9) User asks for advisable actions
(10) Chatbot lists Fire scenario advisable actions
(11) User requests the local fire brigade's contact
(12) Chatbot replies with the contacts for the local fire brigades
(13) User inquires about Alkion's availability
(14) Chatbot replies with disasters affecting Alkion
(15) User queries chatbot for earthquakes that have happened in a certain period
(16) Chatbot replies with information about earthquakes within the detected time-gap

Figure 5.1: Demonstration use-case timeline

5.2 Evaluation

In our work, we have performed three separate iterations of the DSRM process model. All of these iterations had different entry points, which were dependent on the stage of the entire process, as well as on the feedback we have received after each demonstration and evaluation. Figure 5.2 encapsulates and displays all of the iterations we have coursed through.

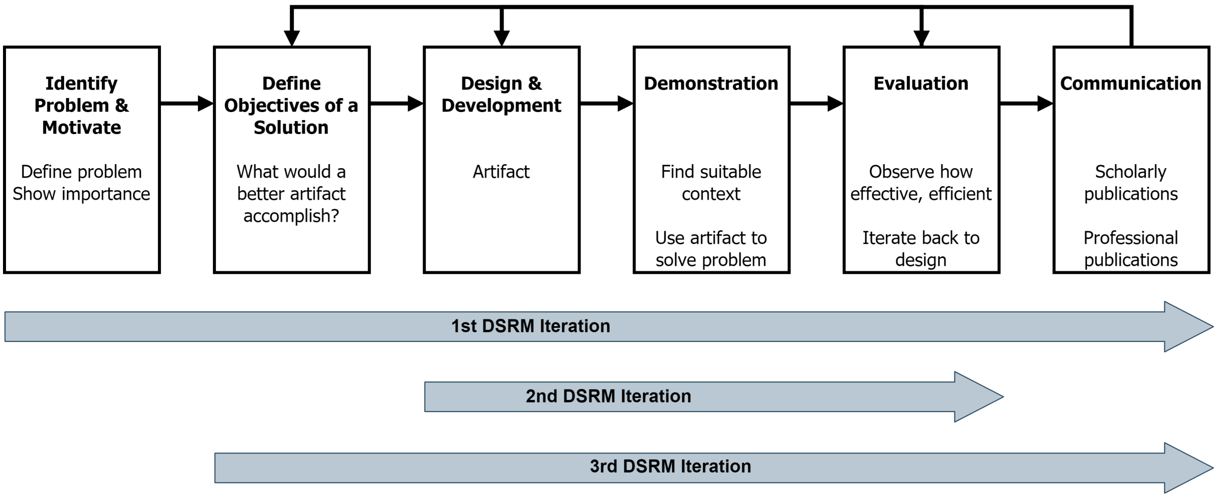

Figure 5.2: DSRM Iterative cycles scheme

5.2.1 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 the artifacts, which has consisted in the great majority of the developed work. This iteration has also included a communication regarding the first artifact, in the form of a peer-reviewed scientific publication (Further information is included in Chapter 6).

The evaluation of this iteration was performed to ensure that our artifacts were fit for purpose, according to the objective statements presented in Chapter 1, that were defined in collaboration with our evaluators. The obtained evaluations are presented in Table 5.1.

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
Performance	Correctly extracting disasters and related information amongst noisy social media environments	PA
Learning capability	Automatically learning about disaster intelligence and updating that knowledge mid disaster	LA

TABLE 5.1: Results of the first DSRM iteration

Due to schedule limitations, for the evaluation of our first iteration only one of the three civil protection specialists we have previously introduced was available. Despite this shortcoming, the session bore interesting results, seeing that although both artifacts have 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 regards of 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, associated with the chatbot. The necessity for a confirmation of the gathered knowledge has also been referred to as an issue. Seeing that the chatbot could easily be disseminating false information to citizens and first-responders, our system could be creating a risks, rather than resolving them.

The knowledge extraction system has also misclassified some disasters, having detected both a flood and an explosion, when they did not belong to the demonstration scenario. Further analysis of this misclassification has shown that even though the detected flood was indeed incorrect, the explosion resulted for tweets which refer explosions resulting from the earthquake, for example "#SismoBarreiro várias explosões ocorridas!/#BarreiroEarthquake several explosions occurred!".

5.2.2 Second DSRM Iteration

The second DSRM iteration was solely focused on fixing all of the issues that had been pointed out in the evaluation of the previous iteration, thus its starting point having been the Design & Development of the artifacts. 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:

- **Integration of the official Google Cloud Services' Translation API** This change has led to an improvement in the quality of the translated tweets, as Google's official API is up-to-date with recent algorithm developments, when compared to the non-official API we had been using before.
- **Disaster extraction Lexicon-based model tweaking** As mentioned before, the poor quality of the available data for training our disaster extraction CNNs has led to the misclassification of some tweets as flood disasters, which led us to implement the Lexicon-based model as a double-check (see Chapter 3 for details). This experimental process has led us to estimating a lower threshold for using the Lexicon-based model when the CNNs classify a disaster as being a flood. Besides this, we have also added the None class to the Lexicon-based model, which is now the output when the lexicon-based model detects no words regarding any of the other classes.
- **Integration of Barreiro's municipality relevant contacts** 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 is 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.

It is also worth mentioning that the demonstration scenario tweets used for testing the artifacts, have not been taken into account when developing these changes. This was essential to avoid grounding our developments in limited data, and consequently developing biased algorithms. Table 5.2 shows the obtained evaluations for this 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
Performance	correctly Extracting disasters and related information amongst noisy social media environments	LA	LA	LA
Learning capability	Automatically learning about disaster intelligence and updating that knowledge mid disaster	LA	LA	LA

Table 5.2: Results of the second DSRM iteration

This iteration's developments 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 reason as to why none of the criteria had yet reached a consensual rating of Fully Achieved (FA), was the limited extent of what the demonstration scenario was showing. In accordance to this, statistics about the disaster detection have been requested. Some minor grammatical errors in the chatbot replies and abrupt dialogue turns have also been reported, along with the inclusion of 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.

5.2.3 Third DSRM Iteration

The third and last DSRM iteration has gone back to the definition of the objectives of our second artifact, the chatbot. 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 this new criteria, and the last iteration concerns regarding the lack of information about the back-end results of the knowledge extraction artifact, the main developments of this iteration were:

- **Chatbot dialogue minor fixes** 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 of the chatbot. 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 chatbot'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.

- **Knowledge extraction results report** We have put together information regarding the back-end results of the demonstration. It includes the final state of the Knowledge Graph, and accuracy of the knowledge extraction.
- **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.

In order to provide the evaluators with more details about the performance of the Knowledge Extraction artifact, we have examined all of the system classifications, and compared them with the expected class we have previously set. In order to achieve this, we have defined misclassification as the artifact detecting the wrong disaster type and/or not detecting the expected entities. Table 5.3 presents the accuracy results for the different sets of tweets.

	Related & Informative	**Related**	**All**
Cases	27	50	800
Misclassifications	5	5	5
Accuracy	81.48%	90.00%	99.38%

TABLE 5.3: Knowledge Extraction accuracy according to tweet predetermined class

We can see that the misclassifications have solely occurred in the Related Informative tweets. Further examination of these tweets led us to conclude that the way our artifact deals with the hashtags is partly responsible for this. It separates hashtags by words starting with a capital letter so, when dealing with lower-cased hashtags, it simply removes them. This has happened to three tweets that only referred the disaster type in a lower-cased hashtag. The remaining two tweets have been directly misclassified by the Disaster Extraction component, which did not detect a relevant disaster due to the lack of direct information, and are as follows:

- O Barreiro está a abanar!/Barreiro is shaking;

- O Barreiro está coberto por uma nuvem de fumo!/Barreiro is covered by a cloud of smoke!

Although their interpretability can be subjective, we have decided to include them in the Related Informative class, and thus have been misclassified by the artifact.

After the first artifact's ingestion of the demonstration's 800 tweets, the Knowledge Base was populated with all the extracted knowledge. Figure 5.3 shows a snippet of the Knowledge Base immediately after this.

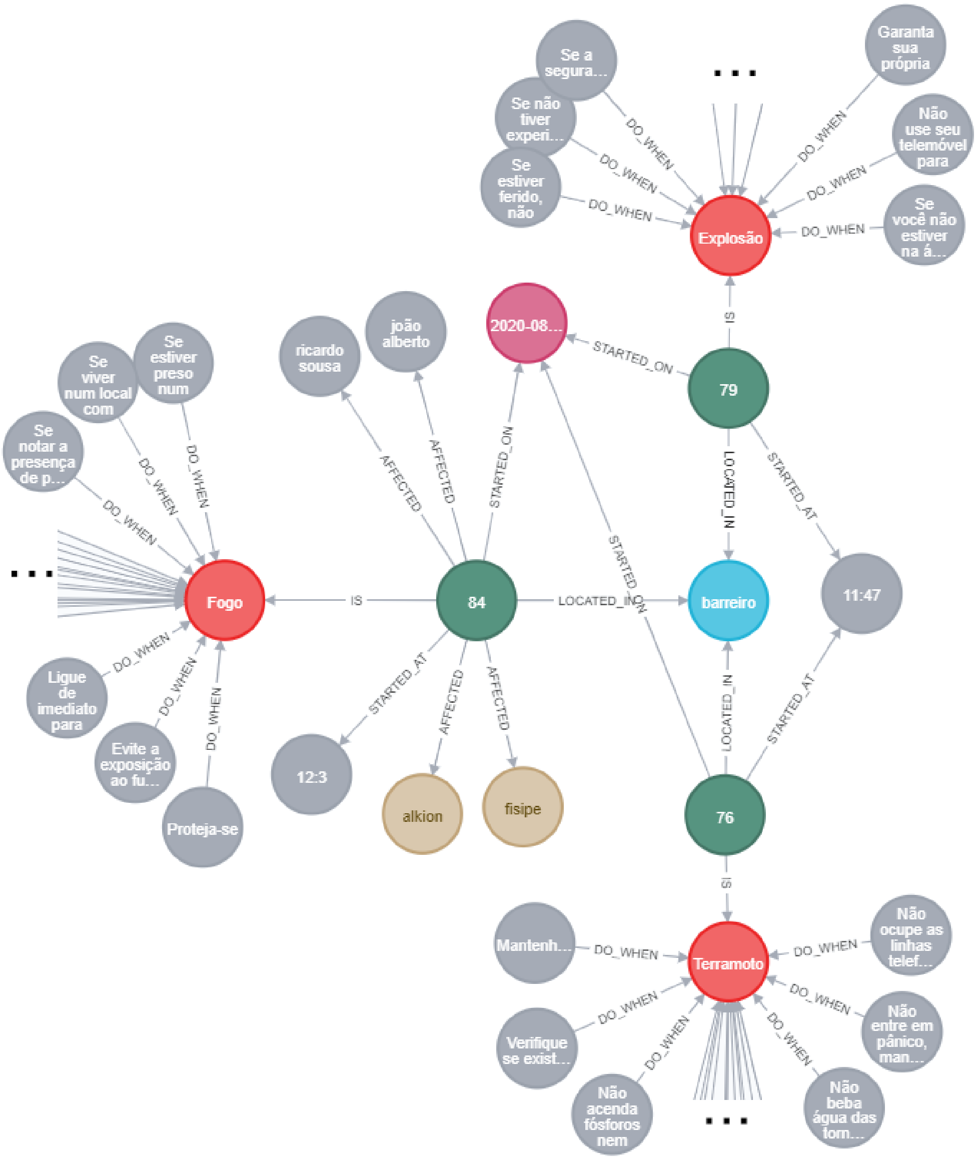

Figure 5.3: Snippet of the Knowledge Base after tweet ingestion

On the account of factors related to the 2020 COVID-19 pandemic, the evaluation of the third iteration of the DSRM process has been postponed to early November 2020. Due to this date's incompatibility with the dissertation's delivery deadline, the evaluation results have not been included in this document.

Despite this shortcoming, we have made an estimation of the results, taking into account previous conversations with the evaluators, and the achieved objectives that have been set for this iteration. Table 5.4 displays the estimated evaluation for the third 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	LA	LA	LA
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	LA	LA	LA
Level of detail	Providing citizens with integrated local knowledge	FA	FA	FA
Performance	correctly Extracting disasters and related information amongst noisy social media environments	LA	LA	LA
Learning capability	Automatically learning about disaster intelligence and updating that knowledge mid disaster	FA	FA	FA

TABLE 5.4: Estimated results of the third DSRM iteration

Chapter 6

Conclusions & Future Work

6.1 Conclusions

The main goal of this dissertation was to devise a disaster-support chatbot system with the capacity to enhance citizens and first responders' resilience in disaster scenarios, by gathering and processing information from crowd-sensing sources, and informing its users with knowledge about the detected threats. With this objective in mind, we have created two innovative artifacts, and two matching main research questions:

- Can enough relevant information be automatically crowd-sourced in disaster scenarios?
- Can a chatbot improve citizens and first responders' informativeness in a disaster scenario?

Throughout our work, we have been able to successfully explore the application of AI's subfields, such as NLP and machine learning, to disaster-management. We have used state-of-the-art technologies, and sometimes created our own, to deal with new and insufficiently addressed tasks. These technologies included neural networks, graph-structured knowledge bases focusing on semantic queries, and recent chatbot developments. We have successfully extracted joint value from these tools to create new and innovative artifacts.

Despite the fact that the results have proven out to positive, many challenges have been surpassed through this process. Leading this work in a symbiotic relationship with a large European project, as well as jointly working with some of its contributors, has proven to be a very rewarding yet difficult challenge. Matching different agendas, has sometimes delayed decisions and created dependencies that have been hard to overcome, but were themselves, part of the challenge we have set ourselves to accomplish.

Although not without its limitations, our first artifact can effectively leverage enough information about threats and surrounding information, especially in crowd-sensing scenarios where the quantity of repeated information being posted makes up for some specific pieces of information that are not retrieved by the artifact's algorithms. This artifact has achieved an accuracy of 99.38% taking into account all of the data used in the demonstration, which is relevant considering that it has not retrieved any false information, and has been able to extract all of the relevant aspects of the scenario. When taking into account the Related Informative tweets only, the system's accuracy decreases to 81.48%, which means that in scenarios with information scarcity, missing this key information could prove to be a problem. Despite needing further work, this artifact has both been approved through peer review, and by field specialists through the DSRM evaluation.

In regards to the second artifact, the chatbot, we have been able to prove that this 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 scenarios. This evidence has also been endorsed by specialists, through the evaluations our artifact has been given, which went from Largely Achieved to Fully Achieved, on all proposed criteria.

As a result of the work performed in this dissertation, and with the guidance of the followed methodology, we have been able to answer the research questions we had initially proposed. As a direct result, we have proven that this novel contribution, the disaster-support dynamic knowledge chatbot, is able to independently extract and present their users with relevant resilience-improving information, and thus being a relevant tool in disaster scenarios. By virtue of surpassing all of these challenges and developing both the back-end extraction and management of the information and one of the main channels of communicating that information, our work has proven out to be one of the pillars of the Portuguese context of H2020 Infrastress. Despite the limited scope of this dissertation, we have had a major role in the development of the project, and will enthusiastically continue working on it until its conclusion, in May 2021.

6.2 Communication

Through this dissertation, different contributions to the scientific community were made:

- Two disaster-management artifacts, validated by field experts and ready for deployment as support tools in disaster scenarios;
- A publication regarding the first artifact (work held in Chapter 3): Boné, J.; Dias, M.; Ferreira, J.C.; Ribeiro, R. DisKnow: A Social-Driven Disaster Support Knowledge Extraction System. *Appl. Sci.* 2020, 10, 6083.;
- (in editorial process) A publication regarding the second artifact (work held in Chapter 4 and the artifact's validation of Chapter 5). This article is named "DisBot: A disaster-support dynamic knowledge chatbot" and is in editorial process for MDPI's special issue *New Directions in Hazard and Disaster Science: Advances in Applied Sciences II.*

6.3 Future Work

Despite the promising results, the work we have carried out in this dissertation can still benefit from future iterations.

As pointed out in the third and last DSRM iteration, the Knowledge Extraction artifact's output is being limited by the way we are currently handling hashtags. Although not optimal, one way to fix this could be implementing a dictionary lookup that tries to match the hashtag's content with word combinations. As for the CNNs we have implemented, their limitations seem to be related with the training datasets that were available at the moment which, although extensive, were only related to a small number of disaster happenings. Future work should include the usage of different training data, and both implementing and tweaking different models. To top this off, the Knowledge Extaction could also benefit from the detection of more and varied entities.

The chatbot has been a focus of constant development, as many user interactions bring out unexpected ways of interacting with the chatbot. We are also aware that it 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 feel unnatural and unexpected.

References

[1] L. K. Reza Farahani Shabnam Rezapour, *Logistics Operations and Management: Concepts and Models.* Elsevier Insights, 2011.

[2] P. L. Bernstein, *Against the Gods: The Remarkable Story of Risk.* Wiley, 1996.

[3] J. Kim and M. Hastak, "Social network analysis: Characteristics of online social networks after a disaster," *International Journal of Information Management*, vol. 38, no. 1, pp. 86–96, 2018.

[4] V. Lanfranchi, S. Mazumdar, E. Blomqvist, and C. Brewster, "Roadmapping discussion summary - Social media and linked data for emergency response," *CEUR Workshop Proceedings*, vol. 1191, 2013.

[5] R. Dale, "The return of the chatbots," *Natural Language Engineering*, vol. 22, no. 5, pp. 811–817, 2016.

[6] J. Weizenbaum, "ELIZA—A Computer Program For the Study of Natural Language Communication Between Man And Machine," *Communications of the ACM*, vol. 26, no. 1, pp. 23–28, 1983.

[7] H. yeung Shum, X. dong He, and D. Li, "From Eliza to XiaoIce: challenges and opportunities with social chatbots," *Frontiers of Information Technology and Electronic Engineering*, vol. 19, no. 1, pp. 10–26, 2018.

[8] M. Dilley, R. S. Chen, U. Deichmann, A. L. Lerner-Lam, M. Arnold, and R. Burby, "Natural Disaster Hotspots: A Global Risk Analysis Synthesis Report," *Earth Science*, vol. 75, p. 132, 2005.

[9] M. Poblet, E. García-Cuesta, and P. Casanovas, "Crowdsourcing tools for disaster management: A review of platforms and methods," in *AI Approaches to the Complexity of Legal Systems.* Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 261–274.

[10] D. Feldman, S. Contreras, B. Karlin, V. Basolo, R. Matthew, B. Sanders, D. Houston, W. Cheung, K. Goodrich, A. Reyes, K. Serrano, J. Schubert, and A. Luke, "Communicating flood risk: Looking back and forward at traditional and social media outlets," *International Journal of Disaster Risk Reduction*, vol. 15, pp. 43–51, 2016.

[11] M. Imran, S. Elbassuoni, C. Castillo, F. Diaz, and P. Meier, "Extracting information nuggets from disaster- Related messages in social media," *ISCRAM 2013 Conference Proceedings - 10th International Conference on Information Systems for Crisis Response and Management*, pp. 791–801, 2013.

[12] D. Jurafsky and J. H. Martin, "Speech and Language Processing, Chapter 25: Advanced Dialog Systems," *Speech and Language Processing*, p. 26, 2018.

[13] K. Peffers, T. Tuunanen, M. A. Rothenberger, and S. Chatterjee, "A design science research methodology for information systems research," *Journal of Management Information Systems*, vol. 24, no. 3, pp. 45–77, 2007.

[14] A. R. Hevner, S. T. March, J. Park, and S. Ram, "Design Sceince in Information Systems," *MIS Quarterly*, vol. 28, no. 1, pp. 75–105, 2004.

[15] M. L. Markus, A. Majchrzak, and L. Gasser, "A design theory for systems that support emergent knowledge processes," *MIS Quarterly: Management Information Systems*, vol. 26, no. 3, pp. 179–212, 2002.

[16] G. Cadete and M. M. da Silva, "Using an enterprise architecture model for assessing the resilience of critical infrastructure," *Safety and Reliability - Safe Societies in a Changing World - Proceedings of the 28th International European Safety and Reliability Conference, ESREL 2018*, pp. 1459–1466, 2018.

[17] N. Prat, I. Comyn-Wattiau, and J. Akoka, “Artifact evaluation in information systems design-science research - A holistic view,” *Proceedings - Pacific Asia Conference on Information Systems, PACIS 2014*, 2014.

[18] “Information technology — Process assessment — Part 2: Performing an assessment,” International Organization for Standardization, Geneva, CH, Standard, 2003.

[19] A. Ahmed, “Use Of Social Media in Disaster Management,” *Thirty Second International Conference on Information Systems, Shanghai 2011*, pp. 1–11, 2011.

[20] E. Steiger, J. P. de Albuquerque, and A. Zipf, “An Advanced Systematic Literature Review on Spatiotemporal Analyses of Twitter Data,” *Transactions in GIS*, vol. 19, no. 6, pp. 809–834, 2015.

[21] B. De Longueville, R. S. Smith, and G. Luraschi, “"OMG, from here, I can see the flames!": A use case of mining location based social networks to acquire spatio-temporal data on forest fires,” *GIS: Proceedings of the ACM International Symposium on Advances in Geographic Information Systems*, no. c, pp. 73–80, 2009.

[22] A. M. MacEachren, A. C. Robinson, a. Jaiswal, S. Pezanowski, a. Savelyev, J. Blanford, and P. Mitra, “Geo-Twitter Analytics: Applications in Crisis Management,” *Proceedings of the 25th International Cartographic Conference*, pp. 1–8, 2010.

[23] D. Murthy and S. A. Longwell, “TWITTER AND DISASTERS: The uses of Twitter during the 2010 Pakistan floods,” *Information Communication and Society*, vol. 16, no. 6, pp. 837–855, 2013.

[24] J. C. Bosley, N. W. Zhao, S. Hill, F. S. Shofer, D. A. Asch, L. B. Becker, and R. M. Merchant, “Decoding twitter: Surveillance and trends for cardiac arrest and resuscitation communication,” *Resuscitation*, vol. 84, no. 2, pp. 206–212, 2013.

[25] A. E. Kim, H. M. Hansen, J. Murphy, A. K. Richards, J. Duke, and J. A. Allen, “Methodological considerations in analyzing twitter data,” *Journal of the National Cancer Institute - Monographs*, no. 47, pp. 140–146, 2013.

[26] T. Sakaki, M. Okazaki, and Y. Matsuo, "Earthquake shakes twitter users: Real-time event detection by social sensors," in *Proceedings of the 19th International Conference on World Wide Web*, ser. WWW '10. New York, NY, USA: Association for Computing Machinery, 2010, p. 851–860.

[27] M. Y. S. Uddin, M. T. Al Amin, H. Le, T. Abdelzaher, B. Szymanski, and T. Nguyen, "On diversifying source selection in social sensing," *9th International Conference on Networked Sensing Systems, INSS 2012 - Conference Proceedings*, 2012.

[28] H. Le, D. Wang, H. Ahmadi, Y. S. Uddin, B. Szymanski, R. Ganti, T. Abdelzaher, O. Fatemieh, H. Wang, J. Pasternack, J. Han, D. Roth, S. Adali, and H. Lei, "Demo: Distilling likely truth from noisy streaming data with Apollo," *SenSys 2011 - Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems*, pp. 417–418, 2011.

[29] J. Rogstadius, M. Vukovic, C. A. Teixeira, V. Kostakos, E. Karapanos, and J. A. Laredo, "CrisisTracker: Crowdsourced social media curation for disaster awareness," *IBM Journal of Research and Development*, vol. 57, no. 5, pp. 1–13, 2013.

[30] F. Atefeh and W. Khreich, "A survey of techniques for event detection in Twitter," *Computational Intelligence*, vol. 31, no. 1, pp. 133–164, 2015.

[31] T. H. Nguyen and R. Grishman, "Event detection and domain adaptation with convolutional neural networks," *ACL-IJCNLP 2015 - 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, Proceedings of the Conference*, vol. 2, pp. 365–371, 2015.

[32] G. Burel, H. Saif, M. Fernandez, and H. Alani, "On Semantics and Deep Learning for Event Detection in Crisis Situations," *Workshop on Semantic Deep Learning (SemDeep)*, 2017.

[33] M. Sit, C. Koylu, and I. Demir, "Identifying disaster-related tweets and their semantic, spatial and temporal context using deep learning, natural language processing and

spatial analysis: a case study of hurricane irma," *International Journal of Digital Earth*, pp. 1–25, 2019.

[34] F. Alam, F. Ofli, and M. Imran, "Crisisdps: Crisis data processing services," in *Proceedings of the 16th International Conference on Information Systems for Crisis Response and Management, València, Spain, 2019*, Z. Franco, J. J. González, and J. H. Canós, Eds. ISCRAM Association, 2019.

[35] Y. Sermet and I. Demir, "An intelligent system on knowledge generation and communication about flooding," *Environmental Modelling and Software*, vol. 108, pp. 51–60, 2018.

[36] M. H. Tsai, J. Y. Chen, and S. C. Kang, "Ask Diana: A keyword-based chatbot system for water-related disaster management," *Water*, vol. 11, pp. 1–19, 2019.

[37] H. Y. Chan and M. H. Tsai, "Question-answering dialogue system for emergency operations," *International Journal of Disaster Risk Reduction*, vol. 41, p. 101313, 2019.

[38] M. Haklay and P. Weber, "OpenStreetMap: User-Generated Street Maps. Pervasive Computing," *IEEE Pervasive Computing*, vol. 7, no. 4, pp. 12–18, 2008.

[39] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, J. Klingner, A. Shah, M. Johnson, X. Liu, Łukasz Kaiser, S. Gouws, Y. Kato, T. Kudo, H. Kazawa, K. Stevens, G. Kurian, N. Patil, W. Wang, C. Young, J. Smith, J. Riesa, A. Rudnick, O. Vinyals, G. Corrado, M. Hughes, and J. Dean, "Google's neural machine translation system: Bridging the gap between human and machine translation," *CoRR*, vol. abs/1609.08144, 2016.

[40] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. Philadelphia, Pennsylvania, USA: Association for Computational Linguistics, 2002, pp. 311–318.

[41] A. Olteanu, C. Castillo, F. Diaz, and S. Vieweg, “CrisisLex: A lexicon for collecting and filtering Microblogged communications in crises,” *Proceedings of the 8th International Conference on Weblogs and Social Media, ICWSM 2014*, pp. 376–385, 2014.

[42] J. Pennington, R. Socher, and C. D. Manning, “GloVe: Global vectors for word representation,” *EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, pp. 1532–1543, 2014.

[43] Y. Kim, “Convolutional neural networks for sentence classification,” *EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, pp. 1746–1751, 2014.

[44] H. He, A. Balakrishnan, M. Eric, and P. Liang, “Learning symmetric collaborative dialogue agents with dynamic knowledge graph embeddings,” *ACL 2017 - 55th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, vol. 1, pp. 1766–1776, 2017.

[45] T. Bunk, D. Varshneya, V. Vlasov, and A. Nichol, “DIET: lightweight language understanding for dialogue systems,” *CoRR*, vol. abs/2004.09936, 2020.

[46] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Advances in Neural Information Processing Systems 30.* Curran Associates, Inc., 2017, pp. 5998–6008.

[47] J. D. Lafferty, A. McCallum, and F. C. N. Pereira, “Conditional random fields: Probabilistic models for segmenting and labeling sequence data,” in *Proceedings of the Eighteenth International Conference on Machine Learning*, ser. ICML ’01. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2001, p. 282–289.

[48] V. Vlasov, J. E. M. Mosig, and A. Nichol, “Dialogue transformers,” *CoRR*, vol. abs/1910.00486, 2019. [Online]. Available: http://arxiv.org/abs/1910.00486

[49] L. Wu, A. Fisch, S. Chopra, K. Adams, A. Bordes, and J. Weston, “StarSpace: Embed all the things!” *32nd AAAI Conference on Artificial Intelligence, AAAI 2018*, pp. 5569–5577, 2018.

Appendix A

Related Tweets

1. EstáTremido As minhas paredes começaram a tremer... é um terramoto no Barreiro?
2. Ocorreu o terramoto de 5.8 na escala de Richter na zona do Barreiro
3. SismoBarreiro Está tudo a abanar aqui!
4. Houve um sismo! IC21 abateu
5. Sismo! Há roturas de água por todo o lado
6. As escolas também foram afetadas. Temos as crianças em panico
7. há perigo de tsunami?
8. SismoBarreiro imagens impressionantes!
9. SismoBarreiro várias explosões ocorridas!
10. O Barreiro está a abanar!
11. RT: @Marco123 Há um incêndio no Barreiro
12. Há montes de chamas e labaredas no Barreiro

13. Chamas intensas no Barreiro!
14. fogo Barreiro margem sul
15. bombeiros sem mãos a medir!
16. FogoNoBarreiro Os tanques da FISIPE estão a arder
17. Estão chamas no telhado da FISIPE Barreiro!
18. fogo fisipe relatos de incêndio com explosão!
19. O incêndio da FISIPE Barreiro espalhou-se para a Alkion
20. Ajuda O incêndio do Barreiro já queimou um cidadão. Acho que foi o Ricardo Sousa

Appendix B

Unrelated Tweets

1. RT @rubennunes724: mandem DM atoa
2. @_mfitas_ é obrigação dela como diretora da escola certificar se que tudo chega ou é tratado com a devida organização e pontualidade
3. @teeh1504 kkkkk mentira ze
4. @leonor_piedade aqui a uns tempos ficas com vontade de lhe partir a boca :)
5. RT @baheazuero: Campanha ForaRoger Todo usando essa foto com a do perfil! Responda o perfil do Bahia come essa foto, subindo Hashtag no…
6. RT @serraa_03: No teu caso o Hugo
7. Qual é a necessidade destes atrasados? NÃO É NÃO. https://t.co/xqlYduwZ5x
8. Sou teamdurex agr, os BR só lamento kkkkkk https://t.co/Wvr6VI9D5m"
9. @catsasilva né, um dia ofende as músicas da queen Taylor no outro dia diz isso"
10. @biarsantos17 Mas o que é que a Prof Lurdes tem a ver com isso? É que pelo que vi foi uma falha da secretaria
11. RT @Joferlanca: @reffoioscunha @YouTube Daqui a anos quando se falar do BB2020Tvi será sempre “aquele BB do Diogo”. Facts are facts.

12. RT @_mariaed_alves: Prazer , Kevin Ps : apaguei pq pensei que ele ia levar a mal , mas n levou tps://t.co/EgnIXBB999"
13. @proddAyyMike devias dar os parabéns
14. @titi63375579 Vou escrever o nome dele no caderno
15. Assim fica tudo mais complicado Começo a ter as minhas dúvidas de que vá durar
16. @daltonceita és bem maluco
17. @grlcurlsss @bufalodrogado O meu preferido e da minha mãe também .
18. Eu odeio tanto esse emoji https://t.co/P7SHXVhvw2
19. eu a fazer o meu retiro espiritual e vcs com esses beefs atoa
20. @RitaBarroso12 Já estou a imaginar, pobre criança

Appendix C

Chatbot Interaction

User input -> Olá bot

Como o posso ajudar hoje?

User input -> está calor aí hoje?

A única onda de calor onde eu vivo é quando treinam os meus algoritmos…

User input -> e por falar em calor, queres almoçar comigo umas sardinhas?

Nem pensar!

User input -> não gostas de comer peixe?

Adoro omeletes the bytes. Mas às vezes dão-me a volta ao sistema!

User input -> e de animais, gostas?

O meu criador incumbiu-me a adoração a texugos.

User input -> nuvens de fumo a surgir pelo ar?

Lamento, não percebi bem. Quis questionar-me sobre ameaças activas neste momento? (Sim/Não)

User input -> Não

Lamento, pode então reformular a sua frase por favor?

User input -> estou a ver um incêndio numa fábrica do Barreiro!

Obrigado por me comunicar essa ocorrência. Infelizmente ainda não tenho capacidade de a registar. Por favor comunique diretamente com o 112 ou a Proteção Civil local (21 206 80 26).

User input -> mas o que devo fazer?

Em caso de Fogo, é aconselhável:

- Evite a exposição ao fumo, tape a boca e o nariz com um pano húmido.
- Se estiver num local fechado, desligue a corrente elétrica e feche o gás antes de abandonar o local.
- Tente apagar o fogo sem colocar a sua vida em risco.
- Se no espaço de um minuto não conseguir apagar o incêndio, deve ligar para os bombeiros.
- Uma vez no exterior, deve aguardar pelos bombeiros e estar pronto para dar indicações/informações que lhe forem pedidas: localização do incêndio, materiais a arder, ações que tomou.

Deseja saber todas as indicações que conheço para cenários de Fogo?

User input -> Não

Okay.

User input -> Tens a certeza sobre esses conselhos?

É isto que penso, mas a minha memória também falha. Por favor confirme a informação.

User input -> obrigado

Sem problema.

User input -> qual é o número dos bombeiros?

Existem vários Bombeiros que poderá contactar:

- Para contactar os Bombeiros Sul Sueste pode usar o 212073032. Alternativamente poderá deslocar-se a Parque Industrial Baia do Tejo.

- Para contactar os Bombeiros Voluntários do Barreiro pode usar o 212073062. Alternativamente poderá deslocar-se a Av. Escola dos Fuzileiros Navais.

User input -> olha, a Alkion ainda está em funcionamento?

Pelo que sei, a organização Alkion está a ser afetada pelas seguintes ameaças:

- Fogo em Barreiro, com início às 12h03 de 11/09/2020.

Devido a estes cenários, a disponibilidade da organização em questão poderá estar limitada, ou mesmo fora de serviço.

User input -> e houve algum terramoto no Barreiro entre dia 3 de Agosto e hoje?

Dentro do espaço temporal que indicou, apenas tenho conhecimento dos seguintes Terramotos:

- Terramoto em Barreiro, com início às 11h47 de 11/09/2020, e estado activo.

O tempo especificado tem início em 2020-08-03 00:00:00 e fim em 2020-09-12 00:00:00

User input -> adeus

Até mais.