

UT Permian Basin

Emergency Chatbot

Andrea Soteldo

COSC-4395

Quan Yuan, Ph. D.

14 December 2022

## Abstract

This paper presents a messaging application implemented in Python that serves as a live chat tool for disaster management. The chatbot was created in response to the floods in several states of Venezuela in October 2022 and the devastating floods in Pakistan in the summer of 2022, highlighting the increasing need for disaster management tools in vulnerable areas that may lack access to resources. The chatbot is designed to provide real-time assistance to users during natural disasters by answering simple questions and connecting them with first responders. While the initial focus of the research is on the efficiency of the chatbot in these tasks, the long-term goal is to incorporate natural language processing (NLP) to support multiple languages and improve the chatbot's ability to understand and respond to more complex queries. The research also examines the potential of chatbots as a useful tool in disaster management by considering its usability, user experience, reliability, and robustness. Overall, this study aims to contribute to the development of effective and accessible disaster management tools that can assist people during times of crisis.

## Table of Contents

## Emergency Chatbot

I.	Introduction.....	4
II.	Design and Implementation.....	5
III.	Challenges.....	11
IV.	Pros and Cons.....	12
V.	Future Research Directions.....	13
VI.	Conclusion.....	17
VII.	References.....	18

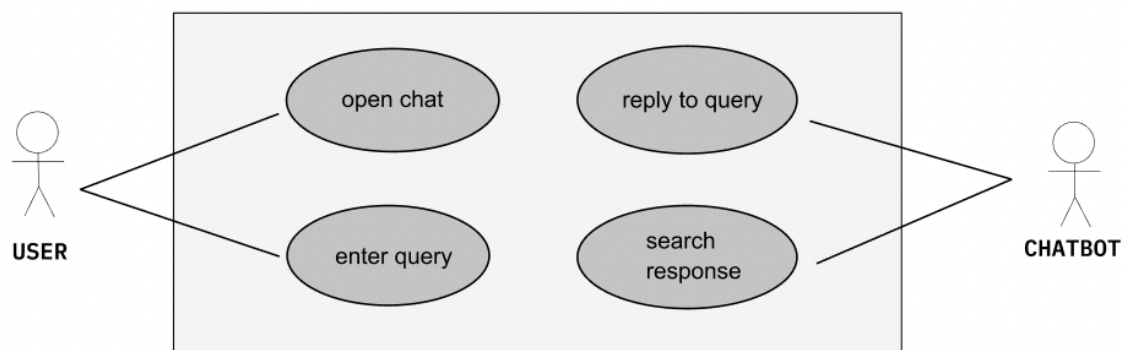
## **Introduction**

This paper presents a chatbot designed to assist people during natural disasters by providing information and connecting them with first responders. The chatbot, called Avila, is trained using a dataset of keywords related to natural disasters and is able to handle tasks such as providing information about shelter, water, food, and medical assistance. The research explores both rule-based and deep learning approaches to training the chatbot, as well as the use of natural language processing (NLP) to improve its performance. The user interface (UI) allows users to access and interact with the chatbot, and the study examines the chatbot's effectiveness in different contexts and its interactions with users. The main focus of the research is on the chatbot's ability to handle simple queries related to the supported tasks, with the goal of understanding how chatbots can be trained to provide effective assistance during times of crisis. The chatbot was created in response to recent floods in Venezuela and Pakistan, highlighting the growing need for disaster management tools in vulnerable areas. Overall, this project aims to contribute to the development of accessible and reliable tools for disaster management.

## Design and Implementation

The chatbot consists of three functional requirements: a json file to store and transmit the data, the main python file to run and train the model, and a User Interface (UI) to expose this data to the user. Therefore, the programming languages explored to develop the application and to achieve these requirements are Python, JavaScript, HTML and CSS. The front-end and back-end of the application were accessed individually to address each feature respectively and then changes were made as deemed necessary.

### Use Case Diagram

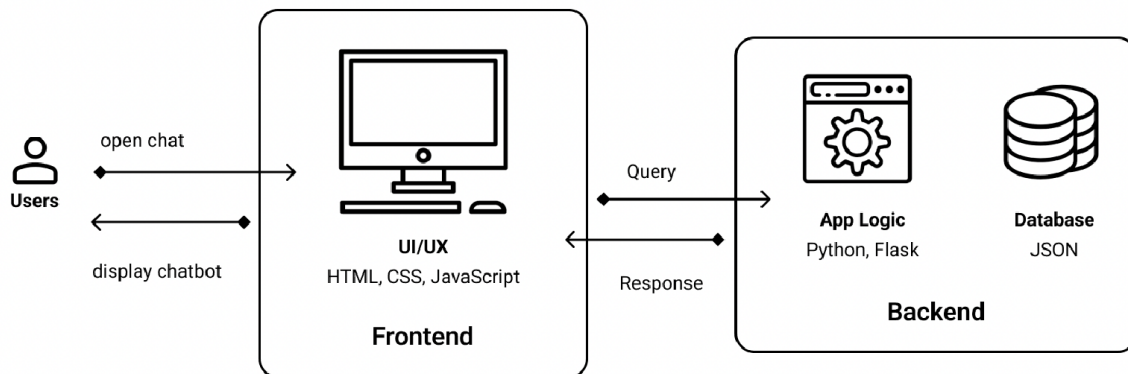


Actor: user

1. The user enters the Emergency Chatbot website.
2. The program displays the landing page and the chatbot can be accessed with the button to the right.
3. The user selects the “chatbot” button.
4. The program displays the “chatbot” GUI.
5. The user inputs the issue and selects the “send” button.
6. The chatbot responds accordingly.

## Architecture Baseline

General description of the application from an architectural view (skeleton of the application).



## User's Manual

Program Overview:

Avila is a simulation of a messaging application that performs live chat functions in response to real-time user interactions during natural disasters. The chatbot works on Windows and macOS, and its operational status is under development.

Program Requirements:

- Windows OS
- macOS

Program Features:

The landing page allows the user to access and interact with the chatbot. The user can request information concerning the following issues:

- Shelter OR electricity
- Water
- Food
- Hygiene products
- Emergency/medical assistance
- Educational resources
- Mental health
- Pet
- Connect with person

Tools:

- Python, JavaScript, HTML and CSS
- Flask

Testing:

Install dependencies (Flask torch torchvision nltk) and run train.py and chat.py

The expected input for the chatbot system is the user's query or request for information or assistance related to the supported tasks during natural disasters. This input is in the form of text, depending on the specific user interface and input methods supported by the system.

The expected output from the chatbot system is a response to the user's query or request, providing the requested information or assistance. This output is in the form of text, depending on the specific user interface and output methods supported by the system. The response should be relevant to the user's query or request, and should be provided in a way that is clear and easy

for the user to understand. The system should also maintain conversation flow and context over multiple interactions, and prioritize the privacy and security of the user's data. If necessary, the system should be able to integrate with other systems to provide the requested information or assistance.

```
What's your full name?
User: Community Member
Hi, Community Member. How can I help you today?
User: I need food for my family
Here's a list of locations with canned goods:
Coming soon.
Is there anything else I can assist you with?[Y/N]
User: y
What else can I assist you with?
User: shelter
Here's a list of locations with reliable shelter:
Coming soon.
Is there anything else I can assist you with?[Y/N]
User: y
What else can I assist you with?
User: emergency
What is the location of the emergency?
User: █
```

Successful criteria:

Success criteria for an emergency chatbot may include the chatbot's ability to provide accurate and relevant responses to user queries, as well as its speed and efficiency in providing assistance. The chatbot's user interface should be user-friendly, especially in potentially stressful or emergency situations. The chatbot should also be able to maintain conversation flow and context over multiple interactions, and have a high level of reliability. Protecting user privacy and security is crucial, as is the chatbot's ability to integrate with other systems as needed. The overall satisfaction of users with the chatbot's performance and usefulness in addressing their needs during natural disasters is also important. Finally, the impact and effectiveness of the



chatbot in assisting people during natural disasters and connecting them with first responders should be measured and evaluated.

### Screen Designs



### Code Sample

Pseudocode:

```
// Feelings=['anxious', 'worried', 'confused', 'helpless', 'hopeless', 'guilty',
'overwhelmed', 'disoriented', 'isolated', 'mental health']

// Need_help=['shelter', 'electricity', 'water', 'food', 'hygiene products', 'medical
assistance', 'emergency', 'educational resources', 'pet']

// Players=['survivors', 'friends', 'loved ones', 'first responders', 'recovery
workers', 'community members']

// Natural_Disaster=['tornadoes', 'hurricanes', 'severe/tropical storms', 'floods',
'wildfire', 'earthquake', 'drought', 'infectious disease outbreak']

// Human_Caused_Disaster=['industrial accidents', 'shootings', 'act of terrorism',
'mass violence', 'community unrest']

// Response_dept=['police', 'ambulance', 'fire brigade', 'paramedics', 'disaster
recovery unit', 'health assistance']
```

```

//Say 'How may I help you?'

//if(users says "shelter"){

// respond: "Here's a list of locations with reliable shelter: "

//}

//if(users says "water"){

// respond: "Here's a list of locations with purified water: "

//}

//if(users says "food"){

// respond: "Here's a list of locations with canned goods: "

//}

//if(users says "hygiene"){

// respond: "Here's the location of sanitation facilities: "

//}

//if(users says "educational resources"){

// respond: "Here's some First Aid training locations: "

// respond: "Here's the government guidelines for preparedness supplies: "

//}

//if(users says "pet"){

// respond: "Do you have a pet?[Y/N]  "

// respond: "Here's some resources: "

//}

//if(users says "emergency"){

// Follow 911 question protocol

// respond: "The emergency response is on the way."

//}

//if(users says [Response_dept]){

// respond: "Redirecting to relevant emergency response department: "

```

```
// respond: "(This is a simulation.)"  
  
//}  
  
//if(users says [Feelings]){  
  
// respond: "Here's some coping tips: "  
  
//}
```

## Challenges

Creating a chatbot can present various challenges, including natural language understanding, conversation flow and context, data availability and quality, integration with other systems, user experience, scalability and reliability, and privacy and security. One specific challenge that was encountered early on in the project was translating pseudocode into functional code that could enable the chatbot, called Avila, to access a library of data and keywords and respond appropriately to users. To overcome this issue, a deeper understanding of natural language processing and chatbot development was needed. Natural language understanding involves the chatbot's ability to comprehend and interpret user input, which can be difficult due to the complexity and diversity of natural language. Conversation flow and context involve the chatbot's ability to follow the progression of a conversation and maintain context throughout multiple interactions, which can be challenging due to the open-ended nature of conversation. Data availability and quality are crucial because chatbots often rely on large amounts of data to learn and improve their performance. Integrating the chatbot with other systems can be difficult due to different APIs and technologies. User experience is important because chatbots should be user-friendly and easy to use. Scalability and reliability are key considerations because chatbots should be able to handle a large number of users and maintain a high level of reliability. Finally, privacy and security are essential because chatbots should ensure the protection of user data.

### Initial Risk Assessment

- Risk 1: Unclear Requirements - Not having a clear understanding of the project requirements can lead to inefficiencies and waste of time. Solution: Good communication skills and thorough research can help prevent misunderstandings and ensure that the team has a shared vision of the desired outcome.
- Risk 2: Poor Scheduling - Poor scheduling can lead to missed deadlines and other conflicts. Solution: Detailed planning and the use of project roadmap tools can help keep the project on track.
- Risk 3: Stretched Resources - Insufficient resources, including time, skills, money, tools, and manpower, can create risk factors. Solution: A resource allocation plan can help make the best use of available resources and support project goals.
- Risk 4: Operational Changes - Unanticipated shifts in responsibilities or procedures can cause distractions and impact project deadlines. Solution: Preparing for and adapting to changes through scheduling tools or additional training can help mitigate this risk.
- Risk 5: System Integration - Integration with the production system can be complex and may involve unexpected barriers due to different software configurations. Solution: Gathering as much information as possible about the configuration early on can help ensure seamless integration.

### **Pros and Cons**

One of the main advantages of chatbots is that they provide a user-friendly interface through conversation, which is accessible to most people. This is in contrast to more complex websites, which may be difficult for some users to navigate. Chatbots can also be delivered

through SMS, which means they can be accessed without an internet connection. This can be particularly useful in emergency situations. However, there are also some drawbacks to chatbots. One limitation is that they often have limited graphical user interface (GUI) options, which can restrict their versatility unless natural language processing (NLP) or deep learning (DL) techniques are used. In addition, chatbots may not always be able to fully understand and interpret user input, leading to potential misunderstandings or incorrect responses. Chatbots may also be less efficient than human customer service representatives in handling complex or nuanced inquiries. Finally, there are concerns about the privacy and security of user data when interacting with chatbots, as well as the potential for chatbots to be used for malicious purposes such as spamming or phishing. Overall, chatbots can provide a convenient and user-friendly interface for certain tasks, but it is important to consider their limitations and potential drawbacks.

### **Future Research Directions**

There are several potential areas of focus for future research on chatbots. One area may be improving the chatbot's natural language understanding capabilities, which could involve researching and implementing advanced natural language processing techniques to better understand and interpret user input. Another potential research direction is enhancing the chatbot's ability to maintain conversation flow and context, which could involve developing algorithms or techniques to better follow the progression of a conversation and maintain context over multiple interactions. Meeting the requirement analysis will also be an important focus, involving identifying the needs and expectations of users and ensuring that the chatbot is able to effectively address those needs. This may include researching and gathering data on user

behaviors and preferences, as well as designing and implementing features and functions that meet user expectations. Other potential areas of focus for meeting the requirement analysis may include addressing scalability and reliability issues, integrating the chatbot with other systems, improving the user experience, and ensuring the privacy and security of user data. By carefully considering the requirements of users and designing the chatbot to effectively meet those requirements, it will be possible to create a chatbot that is useful, user-friendly, and effective.

### Requirements Analysis

The system needs to meet the following functional requirements:

1. Web Interface:

- a. User-friendly conversational interface
- b. Accessible to users
- c. Provides a positive user experience
- d. Allows user to reply
- e. Allows user to share query

2. Backend:

- a. Handles tasks such as providing information about shelter, water, food, and medical assistance
- b. Uses natural language processing techniques to understand and interpret user input
- c. Maintains conversation flow and context over multiple interactions
- d. Able to handle high volume of users and maintain high level of reliability
- e. Prioritizes privacy and security of user data
- f. Able to integrate with other systems as needed

- g. Passes query to chatbot
- h. Searches for query response in dataset

The system may have the following functional requirements:

#### *Undecided*

1. Ability to record ticket/issue
2. Ability to redirect to relevant emergency response department
3. Login and logout from system
4. Admin side with access to record of ticket/issue

#### *Optional*

1. Option for user to attach files/images to chatbot
2. Option for user to send voice messages

Non-functional requirements are unspecified at this point. Foreseeable non-functional requirements may include up-time, reliability, number of concurrent users supported, response time of the system, etc.

#### Questions

- What are the main goals or objectives of the actor in using the system?
  - Receive information and assistance related to the supported tasks during natural disasters
- How does the actor expect to interact with the system (e.g., through a user interface, through voice commands, etc.)?

- User-friendly conversational interface, possibly through typing or voice commands
- What types of queries or requests might the actor make to the system?
  - Information about shelter, water, food, or medical assistance
  - Connection with first responders
- How does the actor expect the system to respond to their queries or requests?
  - Appropriate information or assistance
- What types of information or assistance does the actor expect to receive from the system?
  - Relevant to the actor's queries or requests
- Will the actor need to input any data or information into the system in order to make their queries or requests?
  - Depends on the specific nature of the request
- How does the actor expect the system to handle any errors or issues that may arise during the interaction?
  - Minimizes disruption and ensures actor's needs are still met as much as possible

Additional questions:

- How does the actor expect the system to understand and interpret their queries or requests?
- How does the actor expect the system to maintain conversation flow and context over multiple interactions?
- How does the actor expect the system to prioritize privacy and security of their data?
- How does the actor expect the system to integrate with other systems, if necessary?



## **Conclusion**

The Avila chatbot has been developed to assist people during natural disasters by providing information and connecting them to first responders. It has been trained using a dataset of disaster-related keywords and can handle tasks such as providing information on shelter, water, food, and medical assistance. The chatbot was evaluated using both rule-based and deep learning approaches, as well as natural language processing, and was tested for its effectiveness and user interactions in different contexts. The primary focus of the research was on the chatbot's ability to handle simple queries related to supported tasks, with the goal of understanding how chatbots can be trained to provide effective assistance during crises. In the future, potential areas of focus for chatbot research may include improving natural language understanding, enhancing conversation flow and context, addressing scalability and reliability, integrating with other systems, improving user experience, and ensuring privacy and security. Meeting the requirement analysis, which involves identifying user needs and ensuring the chatbot can effectively address them, will also be important. By considering user requirements and designing the chatbot accordingly, it is possible to create chatbots that are useful, user-friendly, and effective.

## References

- “Chatbot Case Study: BUPA, an International Healthcare Company.” *Tangowork*, <https://tangowork.com/chatbots-case-study-bupa-healthcare/>.
- Adamopoulou, Eleni and Lefteris Moussiades. “Chatbots: History, technology, and applications.” (2020).
- Chaves, Ana Paula and Marco Aurélio Gerosa. “How Should My Chatbot Interact? A Survey on Social Characteristics in Human–Chatbot Interaction Design.” *International Journal of Human–Computer Interaction* 37 (2019): 729 - 758.
- Dahiya, Menal. "A tool of conversation: Chatbot." *International Journal of Computer Sciences and Engineering* 5.5 (2017): 158-161.
- Følstad, Asbjørn and Petter Bae Brandtzaeg. “Users' experiences with chatbots: findings from a questionnaire study.” *Quality and User Experience* 5 (2020): 1-14.
- Ghosh, Shameek, Sammi Bhatia, and Abhi Bhatia. "Quro: facilitating user symptom check using a personalised chatbot-oriented dialogue system." *Stud Health Technol Inform* 252 (2018): 51-56.
- Kohnke, Lucas. "A qualitative exploration of student perspectives of chatbot use during emergency remote teaching." *International Journal of Mobile Learning and Organisation* 16.4 (2022): 475-488.
- Singh, Devanshi et al. “Question Answering Chatbot using Deep Learning with NLP.” *2021 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT)* (2021): 1-6.

Toader, Diana Cezara et al. “The Effect of Social Presence and Chatbot Errors on Trust.” *Sustainability* (2019): n. Pag.

Tsai, Meng-Han et al. “Four-Stage Framework for Implementing a Chatbot System in Disaster Emergency Operation Data Management: A Flood Disaster Management Case Study.” *Ksce Journal of Civil Engineering* (2020): 1-13.

Verma, Siddharth et al. “CHAI: A CHatbot AI for Task-Oriented Dialogue with Offline Reinforcement Learning.” *NAACL* (2022).