5-2022

All The Feels: Building a Chatbot to Assist Students Struggling with Depression

Sara Julieta Tagarot
Trinity University, stagarot@gmail.com

All the Feels: Building a Chatbot to Assist Students Struggling with Depression

Sara Tagarot

Abstract

Inspired by the recent COVID-19 pandemic and its effects on the mental health of millions of people, specifically college students, this thesis focuses on the development of a chatbot to assist those struggling with depression. The bot has three main tasks: identify the user's mood and respond with the appropriate encouraging words, suggest activities that would improve the user's mental health state, and provide resources in case of emergencies.

Acknowledgments

I would like to acknowledge and give my warmest thanks to my supervisor, Dr. Horn. This research would not have been possible if it were not for guidance and patience. I would also like to give special thanks to my partner, my parents, and family as a whole for the continuous unconditional support.

All the Feels: Building a Chatbot to Assist Students Struggling with Depression

Sara Tagarot

A departmental senior thesis submitted to the
Department of Computer Science at Trinity University
in partial fulfillment of the requirements for graduation
with departmental honors.

April 18, 2022

Thesis Advisor   Department Chair

Associate Vice President
for
Academic Affairs

Chapter 1

Introduction

Depression is one of the most serious health problems our society faces as it affects all ages. Ever since the 2020 pandemic 1 every 3 US adults is affected by it[8] and 13.84% of teenagers (age 12-17) have reported in this past year to have suffered at least one major depressive episode[2]. Often times depression is one of the causes for suicide attempts, which is among the main causes of death in the United States and it is on the rise [3] .

In this project we aim to help ease the burden of depression for teenagers and young adults in the United States. Therefore, we will be introducing an intelligent chatbot which can categorize the different moods of the user from the user's responses and provide activities, words of encouragement, and most importantly resources such as hot-lines.

1.1 Chatbot Evolution

The first implementation of chatbots dates back to 1964 with the creation of ELIZA. This bot uses "pattern-matching" and a substitution methodology to provide responses that make the users feel like they are not talking to a bot. The pattern-matching methodology goes through raw data looking for exact patterns and sequences or a sequence of tokens. Eliza, acts a psychotherapist by implementing Rogerian psychotherapy. This means that the bot is interacting with the user showing empathy and lack-of-judgement. The pattern-matching and substitution methodology allows ELIZA to flip the user's sentences into a question to dig deeper into their emotions and thoughts.

Nowadays, chatbots have improved to the point of making ELIZA look obsolete. Many businesses implement various versions of chatbots that provide an immediate answer to users. An example of this is Google's Dialogflow API that can be integrated into different platforms.

Just like the chatbot proposed in this research paper, ELIZA used NLP or Natural Language Processing. This is the branch of Computer Science that helps computers understand text and spoken words similarly to humans. The way Natural Language Processing works, is that once the input from the user is acquired it is split into fragments. These fragments are grammatically analyzed and the meaning of the words are searched. This helps the bot understand the context of the sentence. This steps are repeated automatically with every user response, processing the data in real time. Despite NLP being the backbone of these interactions between humans and computers, there are still many limitations. Language is very complex, with many ways to say the same thing, many different dialects, and it is constantly evolving. The computer must be able to find the meaning of words, analyze a sentence grammatically and semantically, and on top of it all understand its context.

Chapter 2

Background and Related Work

There are a number of chatbots and examples that have been developed throughout the years such as: Eliza, Xiao Ice, Moodnotes, Netomi, etc. Due to this, I will only cover a brief evolution of chatbots and some specific chatbots in health care, along with some key terminology.

The paper by Heung-Yeung Shum, Xiaodong He, and Di Li: " From Eliza to XiaoIce: Challenges and Opportunities with Social Chatbots" goes through the evolution of social chatbots. This includes Task Completion Conversational Agents, Intelligent Personal Assistants (IPA), and Social Chatbots. As well as going through the design principles of chatbots and their frameworks.

Intelligent dialogue systems, also known as social chatbots, have the ability to mimic humans and engage in human conversations. There are also records of chatbots that have passed the famous Turing test, an artificial intelligence method to determine whether a computer is able to think as a human. The first social chatbot to be recorded is Eliza, created by Joseph Weizenbaum at MIT in 1966. Even though this chatbot is unable to actually understand a conversation, it is able to respond appropriately. After Eliza, we come across

Parry, developed by Kenneth Colby (1975) and Alice developed by Richard Wallace (2009). Parry is a chatbot designed to act as a paranoid person would, and it was the first time in history that a computer passed the Turing test. Alice or Artificial Linguistic Internet Computer Entity, on the other hand, uses AI markup language (an XML dialect used to create natural language software agents) and allows users to customize their chatbots.

Moreover, we have intelligent personal assistants such as Siri released by Apple in 2011. This type of system is more complex than the ones above since it can interact with users a lot more; it can perform tasks and services based on questions or commands. These types of chatbots are more task-oriented than conversational chatbots.

Lastly, we get to more complex social chatbots just like XiaoIce developed by Microsoft in 2014. This chatbot is able to understand emotional needs and makes conversation as if it were the user's friend. XiaoIce is able to achieve the main goal of these systems, which is to interact with humans and be their companion while creating an emotional connection [10].

2.1 Chatbots in Healthcare

Thanks to the improvements of chatbots, these are now helping with mental health [9]. They can be considered conversational agents or chatbots since they utilize natural language processing as well as respond using human language automatically. These conversational agents are able to be used through any smartphone and support the user whilst offering them privacy to deal with their mental health concerns. A perfect example of this is Woebot [5], this bot was created in 2017 to monitor its user's mood and engage in therapeutic conversations through Cognitive Behavior Therapy (CBT) among other therapeutic approaches. This type of therapy revolves around the idea that psychological problems are based on learned patterns and the patient in turn can learn strategies to improve it. These therapeutic approaches are then paired up with AI and NLP to converse with the user.

Another similar work is Tobias Bauer, Emre Devrim, Misha Glazunov, William Lopez Jaramillo, Balaganesh Mohan, and Gerasimos Spanakis's research: Building a chatbot to assist survivors of sexual harassment [4]. This non-profit project, through Name Entity Recognition (NER) created a chatbot that can retrieve information from sexual assault survivors. NER can be considered the process the bot undergoes in order to extract information from the user. It specifically seeks to classify entities such as locations or names that are mentioned in unstructured text. In this example, NER is used to find details about the assault, such as the location, the name of the assaulter, etc.

Through the data gathered by SafeCity of harassment in India, the model was trained. The text-preprocessing includes contraction handling, special character removal, spelling correction, negation handling, lemmatization, lower case, and part-of-speech tags. As for text classification, the goal is to find out if the texts are related to a form of harassment issue. In addition, the bot will attempt to extract a detailed description on the harassment by gathering data on location, type of harassment, etc. Based on the data acquired the chatbot will provide the survivor with specific information depending on their case. For example, if physical abuse is detected, the chatbot will advise medical attention and provide a list of providers. Overall, the results were successful with over 80% accuracy.[4].

Chapter 3

Methods

For this project, the platform used is Rasa Open Source (Rasa) [7]. This platform assists in the creation of virtual assistants. Rasa provided a base for the bot that made it the perfect tool for this project due to time constraints.

- Natural Language Understanding
- Dialogue Management

Having a basic structure in place that dealt with the NLP and dialog management aspects opened up extra time to work on the training data. On top of this, Rasa keeps user's conversations completely private and allowed me to deploy the assistant in Trinity's infrastructure. This permits the user to interact with the bot without the need of sending the conversations to a third party. Since the project deals with a sensitive aspect of the users' life this element was key to the development.

3.1 Understanding Rasa

Rasa is used for the development of a task oriented dialogue system. This means that the bot is focused on completing specific tasks such as classifying the mood of the user, providing accurate responses, etc. The training consisted in providing examples that the system learns from in order to generalize patterns in the data.

It comprises two mechanisms: an NLU and Dialogue Policies. The NLU (natural language processing) refers to the part of the system that accepts raw text that is being imputed and converts it into machine readable information. The dialogue policies part of the system predicts the next action to be taken.

The software comes with a neural network architecture called DIET. The DIET network architecture works as a middle-ware. The design of this architecture consists of a *client*, a *master agent* that receives the requests from the user or client, a *local agent* which transmits requests between the master agents and the servers, and lastly a *server daemon* that manages a processor and stores information. DIET sorts texts into intents and entities based on previous examples. This refers to the user's goal or what they are trying to convey, and the keywords that can be found the in the message. [6].

3.2 Identifying the Problems

When it came to installing Rasa into the Trinity University environment, there were some issues that delayed the project significantly. The first problem was due to a large amount of memory needed to download, train, and even run the system. Also, Rasa Open Source needed a modern version of python in order to install. The files needed to successfully install Rasa required more space than the one provided by the machines available which forced going through the university in order to acquire over 3GB of quota space.

Once Rasa was successfully installed and everything seemed to be running properly the environment is scheduled to remove core dump files every night by pattern-matching the keyword “core”. This deleted files that were needed in order for Rasa to function including: TensorFlow and numpy since it overlapped with their “core” libraries. In order to correct this, every morning after the files were erased they would be re-installed for the day.

However, the biggest obstacle to tackle was training Rasa in order to be able to carry out a normal conversation without triggering the user. The bot had to pass several trial conversations before being used on real users that experience depression. Once it was able to carry out a conversation that would not potentially stress the user, a controlled environment was used to carry out these interactions.

3.3 Training Data

The training of the AI assistant was done through the command line in a virtual environment. Shortly after initializing Rasa you can train the model by executing the “Rasa train”. However, since the assistant is built to help and provide resources for people struggling with their mental health, the training data and actions had to be customized.

Rasa uses YAML [1], an Unicode based data serialization language, to manage all training data. The domain, or the inputs and outputs of the assistant, utilize the same YAML format as the training data. There are a number of available “keys” that can be used only once in a single file, such as: version, nlu, stories, and rules. For example, the *version* key must be specified at the beginning of every file; otherwise, the bot assumes you are using the latest format specification that is supported. This chatbot uses the 2.0 version. The other keys are split into different yml file that will be explained in the next number of paragraphs.

The NLU training data contains several examples of user utterances categorized by intents. For example, figure 3.1 shows training data for when the user is not having a good day and labeling this intent “mood-awful”. All training examples are grouped by the user’s intent and listed under the examples key. In this case the intent is labeled “mood-awful” but some other intents that can be found in this bot are mood-unhappy, mood-great, deny, etc. The possible sentences that might trigger the assistant to recognize this intent are the examples listed below the intent. These user intents can be found in the nlu file.

```
- intent: mood_awful
  examples: |
    - I don't want to feel anything anymore
    - I have no motivayion anymore
    - awful
    - horrible
    - terrible
    - I am doing terrrible
    - life sucks
    - this sucks
    - I hate my life
    - I hate this
    - I hate everything and everyone
    - I need help
    - I am not doing well today
    - I am having a breakdown
    - I am breaking down
    - I'm done
    - I'm done with this
```

Figure 3.1: nlu.yml

Another type of training data is the Conversation data. This data includes rules and stories, which are used to train the dialogue management model. Rules describe small pieces of conversations that follow a fixed behavior, meaning they go through the same path no matter what. These are also used to train the Rule-Policy. The Rule Policy is the part of the bot that handles the conversational parts that follow a fixed behaviour and makes predictions on what the bot should respond. On the other hand, stories train a machine learning model to identify conversational patterns.

For this assistant, the files that contain customized training data are: domain.yml, nlu.yml, rules.yml, and stories.yml.

- NLU file contains the user utterances categorized by intents. In order for the assistant to properly respond to the user's emotional state, the training data for the moods was split into three different categories or intents: mood-great, mood-unhappy, and mood-awful. The examples for each of the mood intents was gathered from day to day conversations held with Trinity University's students.

- Stories file contains different paths of a conversation between the user and the assistant. These include the expected reactions of the assistant to the different moods of the user. In fig 3.3 where the story is labeled "sad path 3" we can see that the conversation between the assistant and an user that is having a terrible day would go from start to finish.

- Domain file contains the different responses the assistant can carry out. Some of the customizations include offering different encouraging words depending on the mood of the user, information on hot-lines and other resources, and activity suggestions to help improve the user's mood.

- The rules file handles small conversational patterns. In this case the file has three separate rules: one in case the bot is ever asked whether it is human or a machine, another to say goodbye whenever the user says goodbye, and lastly to say you are welcome and ask the user to check in tomorrow whenever the user is satisfied and says thanks. The bot already came with the bot-challenge to add functionality.

3.4 Chatbot

Whenever the chatbot greets the user it asks how you are feeling. Based on the user's response, the bot can label the emotion as happy, unhappy, or awful. There are different possible paths the bot can take once the mood has been detected. Most commonly, the chatbot will reply with some words of encouragement and ask if the user would like activity suggestions that have shown to improve mental health based on the HelpGuide's 6 keys to mental health: social connection, physical activity, managing stress, healthy diet, quality sleep, and meaning/purpose. In the case that the bot recognizes the user's mood to be awful, one of the first things will be to send one of the many resources for suicide prevention the bot has stored in the domain.yml file.

```yaml
- intent: mood_great
  examples: |
    - perfect
    - great
    - amazing
    - feeling like a king
    - wonderful
    - I am feeling very good
    - I am great
    - I am okay
    - It is not a bad day
    - I am amazing
    - I am going to save the world
    - super stoked
    - extremely good
    - so so perfect
    - so good
    - so perfect
- intent: mood_unhappy
  examples: |
    - I am sad
    - I don't feel very well
    - I am disappointed
    - I'm so sad
    - sad
    - very sad
    - unhappy
    - not good
    - not very good
    - so saad
    - so sad
    - I'm starting to feel bad
    - a little sad
    - I am having a day
    - I have little motivation today
    - It is not the best day
    - I'm so stressed
    - stressed
    - I could be doing better
    - I could be doing worse
- intent: mood_awful
  examples: |
    - I don't want to feel anything anymore
    - I have no motivayion anymore
    - awful
    - horrible
    - terrible
    - I am doing terrrible
    - life sucks
    - this sucks
    - I hate my life
    - I hate this
    - I hate everything and everyone
    - I need help
    - I am not doing well today
    - I am having a breakdown
    - I am breaking down
    - I'm done
    - I'm done with this
```

Figure 3.2: nlu.yml

```yaml
- story: sad path 3
  steps:
  - intent: greet
  - action: utter_greet
  - intent: mood_awful
  - action: utter_help2
  - action: utter_resources
  - action: utter_want_suggestions
  - intent: affirm
  - action: utter_suggestions
```

Figure 3.3: stories.yml

```yaml
  utter_resources:
  - text: These might help Suicide Prevention Lifelines - 8002738255 or 1800
2738255
  - text: This is the 24/7 Crisis Hotline 1800273-TALK (8255)
  - text: If you are not feeling well at all you can always call 911 for hel
p.
```

Figure 3.4: domain.yml

```yaml
- rule: Say goodbye anytime the user says goodbye
  steps:
  - intent: goodbye
  - action: utter_goodbye

- rule: Say you are welcome anytime user says thanks
  steps:
  - intent: thanks
  - action: utter_welcome
  - action: utter_check_in_tomorrow
  - action: utter_goodbye

- rule: Say 'I am a bot' anytime the user challenges
  steps:
  - intent: bot_challenge
  - action: utter_iamabot
```

Figure 3.5: rules.yml

Chapter 4

Results

Once the bot was trained and ready to interact with real users, 5 volunteers were asked to interact with it and then complete a questionnaire to assess the efficiency of the chatbot overall. The most relevant question was answering how they felt from 1-10, 1 being awful and 10 being great. The volunteers answered this question before and after the interacted with the bot. Each user interacted with the bot reflecting one of the three different mood categories. Two users were 'unhappy' with a rate before they interacted with the bot of 4 and 5. Two users stated that they felt 'great' with a rate of 8 before the interaction. Lastly, one user fell under the 'awful' category with a rate of 2 before interacting with the bot. Some of the 'unhappy' and 'great' mood conversations can be found in figure 4.1.
The questions in the questionnaire were:

- Q1. How are you feeling today from 1-10? (1 being awful and 10 great)
- Q2. How are you feeling after chatting with the bot? (1 being awful and 10 great)
- Q3. What did you NOT like about the assistant?

- Q4. What would you like the assistant to have/say/do that it did not?
- Q5. How likely are you to use the chatbot again? (1 being never and 10 tomorrow)

Altogether, every user experienced an improvement of one or more points higher than their before rate. The one user that fell into the 'awful' category showed a 2 point improvement after interacting with the bot. On a later check-in with that same user, they stated that after doing the activity suggested by the bot (journaling and meditation), they would rate their mood as a solid 6.5. This shows an overall 4.5pt improvement in their mood after interacting with the bot.

Whilst the questionnaires were an effective way to gather the emotional state of the users before and after the interaction, it may be subject to bias. This is the case since it does not take into account the environment the user was in before nor after the questionnaire was filled out. It also does not take into account whether the user had experienced something out of the norm that day that may have affected their mood.

```
Your input ->  hey
Hey! How are you?
Your input ->  not so good
You are not alone.
Do you want some ideas for activities that might cheer you up?
Your input ->  sure
Maybe it's time to journal, try meditation, or even some yoga!
Did that help you?
Your input ->  yeah that helps, thanks!
You are welcome!
Thanks for checking in today!
Bye
```

```
Your input ->  Hello
Hey! How are you?
Your input ->  Could be better
You got this, just one task at a time.
Do you want some ideas for activities that might cheer you up?
Your input ->  yeah
Is there maybe a book you like to read?
Did that help you?
Your input ->  not really
If you are not feeling well at all you can always call 911 for help.
Your input ->  thank you
No problem!
Can't wait to talk to you tomorrow!
```

```
Your input ->  Hello
Hey! How are you?
Your input ->  Doing okay
yay! Let's keep the good vibes going!
```

Figure 4.1: Results

Chapter 5

Discussion

Overall, all the users showed that they felt better after interacting with the chatbot. In addition, all the users mentioned in the questionnaire that the activity suggestions were their favorite aspect. Those users that tried the activities the bot suggested reached out saying that “it definitely helped release stress” and they were “so glad they did the activity cause it was very helpful”.

On the other hand, 3 of the users stated they wished the chatbot would be more accessible. In the same questionnaire the users stated they wished they were able to check in at any time of the day and any location.

Chapter 6

Conclusions and Future Work

Some additions that the original model could go under in the future would be:

- Adding the model to a mobile app to make it accessible to a wider audience. Based on the user's feedback, the ability to check in with the chatbot at any time and place is crucial for avoiding and/or managing depressive episodes.

- In addition to this, adding a 'panic' command or button within the bot would be the next step. Whilst Rasa does not offer this type of functionality yet, the button would have to be added after the chatbot integration with a website or app. This panic command would allow the user to automatically connect to a therapist or someone trained to assist them in case of an emergency.

Other than these, there are multiple improvements the bot could undergo. Some other ideas involve: allowing the bot to discuss stressful events with the user, being able to track their mood check-ins in a visually pleasing way, and ultimately be able to implement different psychotherapy approaches to help the user deal with depression, specifically Rogerian Psychotherapy. This would ultimately require a lot more training data to be added, and the integration to be successful. A possible route would be upgrading to a paid version of Rasa instead of using the free version (Rasa Open Source). This would allow the training data to be automated and not have to be manually inputted.

Bibliography

[1] Yaml Markup Language revision 1.2.2.

[2] Youth Data 2021. 2021.

[3] Suicide. 2022.

[4] Tobias Bauer, Emre Devrim, Misha Glazunov, William Lopez Jaramillo, Balaganesh Mohan, and Gerasimos Spanakis. #MeTooMaastricht: Building a chatbot to assist survivors of sexual harassment. *arXiv:1909.02809 [cs]*, September 2019. arXiv: 1909.02809.

[5] Woebot Health. Woebot Health. Relational Agent for Mental Health. 2022.

[6] m vdb. Introducing DIET: state-of-the-art architecture that outperforms fine-tuning bert and is 6x faster to train. 2021.

[7] m vdb. Introduction to Rasa Open Source. 2021.

[8] Jillian McKoy. Depression Rates in US Tripled When the Pandemic First hit - Now, They're Even Worse. 2021.

[9] Sohail Merchant. These Chatbots are Helping with Mental Health right now. 2021.

[10] Heung-yeung Shum, Xiao-dong He, and Di Li. From Eliza to XiaoIce: challenges and opportunities with social chatbots. *Frontiers of Information Technology & Electronic Engineering*, 19(1):10–26, January 2018.