Quinn: Medical Assistant for Mental Counseling using Rasa Stack

Sandra V A[1], Vinitha V[2]

[1]*Master of Computer Applications, College of Engineering, Trivandrum, Kerala, India*
[2]*Asst. Professor, Dept. of Computer Applications, College of Engineering, Trivandrum, Kerala, India*

---***---

Abstract - *Many patients undergo psychotherapy due to changes in awareness of psychiatric treatment. Despite many mental illnesses in modern people, the real rate of diagnosis and treatment is still low. Compared to demand, there is a lack of experts and even the cost of consulting the experts is high. A chatbot called Quinn is developed to serve the users as a personal assistant to solve the problem. It is a chat robot that interacts with the user to analyze their thoughts or state of mind and suggests appropriate solutions. Thus, the conversational service can provide individuals with personalized counseling. One-to-one conversation can effectively resolve the isolation. Quinn notifies the users' dangerous status when there is an accidental mental disorder, such as panic and suicidal impulse. Besides, the system observes the mood swings continuously for users who are manic-depressive. This conversational service for psychiatric counseling adapts methodologies to understand counseling contents based on high-level natural language understanding (NLU) using Rasa Stack.*

Key Words: Medical Assistant, Chatbot, Mental Counseling, Natural Language Understanding(NLU), Rasa Stack, Rasa NLU, Rasa Core

1. INTRODUCTION

Over the past few years, there have been reports revealing that most of the people suffer from some form of mental illness. This is because as the world advances to the next level, it expects better technological as well as intellectual solutions from humans for its smooth running. These expectations create a huge pressure that affects people mentally. This is where artificially intelligent chatbots come into the limelight. Chatbots are artificially intelligent robots that take up the conventional tasks which are otherwise done by human beings. Nowadays we can find an array of websites using chatbots to facilitate product sale and customer feedback. Once a chatbot is deployed to a domain, it completes its task without help from humans as they are trained efficiently to retrieve information from its users just like the humans do. Moreover, bots offer help, support, and companionship to its users. But there is one area in which none of these are used, and that is, treating patients.

Depression is a psychological illness that happens due to hormonal changes and factors like variable blood sugar levels. It is difficult to understand if a person is suffering from depression or not. It is a mental illness that requires care and there may be situations where patients want someone to listen to them without being judged. The system "Quinn" is a chat robot developed to provide companionship to those who suffer from depression. The users can chat with the bot and get replies based on their messages.

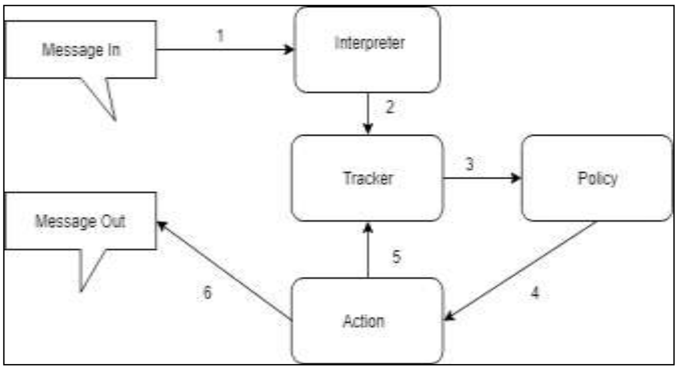
This chatbot is developed with the help of Rasa Stack that offers a set of open source learning tools to create contextual chatbots and helpers with text or voices. It consists of NLU which is an open-source natural language processing tool for intent classification and entity extraction in chatbots[1]. Quinn understands the meaning of the user input based on intent classification and extracts entities if there are any. The Rasa core decides what happens next in this conversation. It is a machine learning-based dialogue management that predicts the next best action based on the input from NLU, the conversation history, and the training data. Apart from chatting and giving appropriate replies, Quinn also provides an option to efficiently book an appointment with the concerned doctor on the date entity extracted from the user's message. It notifies the doctor via email when the user expresses any kind of suicidal thoughts. Thus Quinn is more than a chatbot and hence it can be considered as a medical assistant.

2. PROPOSED SYSTEM

Quinn is an artificially intelligent chatbot that is used to act as a mentor to the user. It works in a closed domain that involves mental counseling for depressed users. The bot understands the user input and gives an appropriate reply to the user, thus giving him/her a feeling of talking to a human being. This functionality is achieved with the help of Natural Language Understanding provided by Rasa. It uses intent classification and entity extraction. Intents are defined with large amounts of data and are trained to create a training dataset. Bot utterances and custom actions are also defined. Stories are created to specify the dialogue flow.

The process of incoming messages is split into different components. These components are executed one after another in a so-called processing pipeline. There are components for entity extraction, for intent classification and pre-processing. The system uses spacy_sklearn for intent classification and entity extraction. This defines the machine learning model. The model is created and trained with the help of trained data, stories and bot utterances. Rasa provides an interactive learning platform that allows chatting with the bot, thus providing a form of reinforcement learning. This asks the user if the classification of the bot is correct or not and create new stories by adding appropriate actions for the classified intent. Once the model is created, the rasa core server is run which can be connected to a UI.

2.1 Architecture



First of all, the message from the user is received and passed to an Interpreter that converts it into a dictionary including the original text, the intent, and any entities that were found. It then passes to Tracker which is an object that saves the dialogue state. There is one tracker object for each conversation session and this refers to the only stateful component in the system. It keeps track of all the slots and log of all the two events that led to that state and have occurred within a conversation. The state of a conversation can be reconstructed by replaying all of the events.

Thirdly, the policy receives the current state of the tracker. The job of a policy is to select the next action to execute given the tracker object. A policy is instantiated along with a featurizer that creates a vector representation of the current dialogue state given the tracker. The standard featurizer includes all the features describing what the last action was, the intent and entities in the most recent user message and which slots are currently defined. The featurization of a slot may vary. There is a hyperparameter max_history which specifies the number of previous states to include in the featurisation.

The fourth step indicates that it is the policy that chooses which action to take next. The chosen action is logged by the tracker which is the fifth step. Finally, the action is executed (this may include sending a message to the user). If the predicted action is not 'listen', the flow goes to the third step [2].

2.2 Database design

XAMPP is used to start the MySql server where the booking details are stored. A table named 'appointment' is created that consists of patient name, appointment date and token number. It validates the input date and then the appointment is confirmed. A token number below 10 is generated only if the user has not placed a booking request on the date which already has an appointment for the user. Only 10 appointments can be made on a particular day. The negative thoughts, as well as the health issues of the user, are stored in the table called health. The medical report is generated with data in this table which consists of patient name, problem and date time.

3. IMPLEMENTATION

3.1 Algorithm

Algorithm for Quinn is as follows:

```
1: Creation of data with multiple intents
2: Training of data
3: Testing with test data
4: Creation of bot utterances
5: Creation of stories
6: Creation of nlu model
7: Adding data and stories using interactive
learning
8: Creation of Dialogue Management model
9: Running custom action server
10: Running the rasa core server
11: Connecting the server with UI
```

The implementation of Quinn involves the creation of two models which are the NLU model, as well as, the Dialogue management model. A bot cannot understand the unstructured human language. Hence it is necessary to convert the high level and unorganized human language to a structured form that is understandable and distinguishable by the bot. This can be achieved by training an NLU model that takes training data with all the data concerned with a particular intent listed under that intent.

A large number of data can be added under each intent to increase the efficiency of the intent classification. One can also specify the information that needs to be extracted and the name of the slot to which the extracted information is stored. Synonyms, as well as, regex patterns can also be specified in the data file. Thus the nlu data file consists of user messages and its intents with entities to be extracted. After the nlu model has been trained, a test sentence can be provided to check if it performs the intent classification accurately

The dialogue management model is a machine learning model that is trained to predict "what the bot should do next" based on the context and state of the conversation. It replaces hardcoded conversation rules. To create a dialogue management model, a domain file, as well as a training data file, is required. A domain file defines the environment where the bot works. It includes the declaration of the slots, entities, intents, and actions as well as defines templates. Templates are nothing but bot utterances. The training data file is a series of actual conversations that the user has with the bot in the form of stories.

To train the dialogue management model, an Agent class that takes a domain file and policies as parameters are used. MemoizationPolicy, KerasPolicy, FormPolicy and FallbackPolicy are the policies used. Memoization policy is used to remember the events as per story and KerasPolicy is used to predict the next action if the user asks something different from stories [3]. FallbackPolicy checks whether the Rasa Stack component Rasa NLU responsible for understanding user messages was able to classify the user message confidently. If the classification confidence is below a certain threshold, it triggers a fallback action[4]. FormPolicy is specified to implement slot filling in an effectively. It extracts information from the user before any utterance or action is executed. The model is thus trained with the help of agent with all the sufficient parameter and it gets persisted in the specified model path.

Custom actions are defined in python. Quinn extracts the name of the user and greets by his/her name. It sends mail to the doctor when the user expresses any suicidal thoughts. It also assists in booking an appointment with the doctor. All these are custom actions are made available by running the custom action server.

Now, the rasa core server is run. For this, an agent class is required which takes the dialogue management model as domain, nlu model as an interpreter and an action endpoint. Action endpoint is url where the trained model is loaded and ready to work as a chatbot. An input channel is created with a port number that can be connected to the designed user interface. The bot can in parallel be integrated with Google Assistant just by adding another input channel with another port number. But since the Google Assistant is an internet service and the bot resides in the localhost, it requires 'ngrok' that acts as a tunnel between the bot and the Google Assistant. ngrok is run at the port number specified in the input channel which was defined earlier. This way one can connect the trained bot to various platforms like Slack, Messenger, Twilio, etc.

4. ADVANTAGES

The system consists of several advantages. Chatbots are having a significant impact on numerous fields especially the psychology sector. Developers caution these tech tools aren't a replacement for human interactions with experts, but it is already clear that chatbots are an always-available resource which isn't the case for human health practitioners. Even when individuals have access to mental health assistance in their areas, they may delay taking advantage of the services for many reasons. They might fear judgment from therapists or people they know. Or, individuals may assume everyone goes through the emotions they're experiencing, and think their feelings aren't severe enough to warrant treatment. Quinn could reduce avoidant behaviors that could cause a downturn in mental health. It can detect if the user is feeling down or not from the message. In that case, Quinn recommends the user to seek professional help, depending on the results.

5. CONCLUSION AND FUTURE SCOPE

Chatbots are applications that are still in the initial level of development. Psychiatric counseling is one area where chatbots are not much utilized. The system "Quinn" is an artificially intelligent robot that would be useful in the psychiatric sector. It is developed to provide basic mental counseling for depressed patients, thus giving them a feeling that there is someone who listens without judging them. Apart from its basic function, it allows patients to book appointments. It automatically sends mail to the concerned doctor in case if there are any suicidal impulses from the patient. Reinforcement learning is also implemented to some extent using the concept of entity extraction or slot filling in Rasa. The system successfully does the above functionalities with appreciable accuracy.

However, a large amount of data is required to train the bot as the efficiency of the bot is directly proportional to the training data. One cannot predict what the user might say and so a large amount of data is required. Apart from that, the developer needs to train the model with the dialogue flow. Moreover, the training data, as well as the stories, are manually created, unlike the ordinary machine learning system where data can be downloaded from external sources. Quinn can classify the input and execute the appropriate action but there might be a message that is random or out of its domain. This is where its efficiency is reduced.

The system Quinn can be extended in such a way that answers all types of queries of the user and provide appropriate solutions. Live reinforcement is one attractive factor that can be added to it. For example, the bot can ask the user what it should do when it cannot answer the random or out of domain question of the user and provide this answer the next time he/she asks the same query.

More and more custom actions can be added so that it increases Quinn's functionality.

REFERENCES

[1] Rasa.com. (2019). Getting Started with Rasa. [online] Available at: http://rasa.com/docs/getting-started/ [Accessed 30 Apr. 2019].

[2] Bocklisch, T., Faulkner, J., Pawlowski, N. and Nichol, A. (2017). [online] Alborz-geramifard.com. Available at: http://alborz-geramifard.com/workshops/nips17-Conversational-AI/Papers/17nipsw-cai-rasa.pdf [Accessed 30 Apr. 2019].

[3] Rasa Community Forum. (n.d.). Memoization Policy with max history. [online] Available at: https://forum.rasa.com/t/memoization-policy-with-max-history/2021 [Accessed 30 Apr. 2019].

[4] Medium. (n.d.). Failing Gracefully with Rasa. [online] Available at: https://medium.com/rasa-blog/failing-gracefully-with-rasa-8ead6b43f2f4 [Accessed 30 Apr. 2019].