

Advanced Healthcare Chat Bot using Python

Raghu Ram Vadlamudi
Student, B.E-(IoT)
Department of CSE,
Chandigarh University,
Chandigarh, India
19BCS4596@cuchd.in

Ranjit Banerjee
Student, B.E-(IoT)
Department of CSE,
Chandigarh University,
Chandigarh, India
19BCS4619@cuchd.in

Harsh Raj
Student, B.E-(IoT)
Department of CSE,
Chandigarh University,
Chandigarh, India
19BCS4586@cuchd.in

Arthi Reddy Dharma
Student, B.E-(IoT)
Department of CSE,
Chandigarh University,
Chandigarh, India
19BCS4593@cuchd.in

+

Abstract— In today's world, health care has become a very important part of our existence. However, it has become very difficult to get a consultation with a doctor about every health-related issue. The idea of this project is to develop an artificial intelligence-based medical chatbot to assist with diagnosis of the problem. Specifically, the project aims to develop a medical chatbot based on artificial intelligence that helps diagnose a health issue and gives an overview of the disease we have and refers us to a doctor for further guidance. As a result of this project, the healthcare cost will be reduced since there are no longer need to see a doctor for minor health problems, and as a result the bot will also perform better since it takes advantage of Natural Language Processing (NLP) to interact with the user, and it stores the data. Using background data, the chatbot can improve itself in order to provide a faster resolution for future questions You can install the chatbot anywhere, such as on a web page or in an app, depending on your needs

Keywords—Chatbot, HealthCare, Natural Language Processing

I. INTRODUCTION

AI software is used to build chatbots, which simulate conversations (or chats) with users through messaging apps, websites, mobile apps, and phones utilizing natural language.

This application is fundamental since knowledge is stored in advance and as a result, it can learn itself and restore its knowledge via human assistance or web resources. As the system application is in the form of a chatbot, it uses the question-and-answer protocol in order to respond to the users' requests.

This system is developed to reduce healthcare costs and time for the users since they are not able to visit a physician or an expert when they need them immediately. As the user asks questions, the chat bot can analyse them so that it can provide the best possible answer to that question. I also have a method of storing previous user data so that I can take their feedback when they return. For the bot to learn from the old cases so it can improve for future cases, we need feedback(Optional)from the user whether the given previous solution is correct

The question is how to make Chatbot system smart enough. There are two ways of doing so:

- **Rule-based Chatbots:** Here a Chatbot system works on basis of certain rules. However when the input pattern does not match with any predefined rule then this Chatbot system is inefficient to answer the question. Developers use AIML (Artificial Intelligence Markup Language) to write rules for Chatbot System. AIML is XML based language. Composing rules for various situations is a very tedious job and it is difficult to write rules for every possible situation. These rule-based Chatbot systems can deal with straightforward questions but it are crucial to managing complex questions. Most social media Chatbot systems are rule-based.

- **Self-learning Chatbots:** These Chatbots use machine learning algorithms that enable them to learn things. These bots can be of two types: Using Retrieval Based Models: These bots are trained for a lot of inquiries and their possible answers. For each question, the bot can locate the most important answers from the set of every conceivable answer. Likewise, there is no issue with the language and sentence structure as the appropriate responses are pre-decided and it can't turn out badly in a sentence structure way. Using Generative Models: Generative models do not reply with the same answer from a set of answers. They take word by word from the inquiry and give appropriate responses. These models should be prepared more exactly because they can handle spelling and grammar errors easily.

Self-learning chatbots are not yet mature. Often, these chatbots provide irrelevant answers to user queries, which can create an irritating experience for business customers. That's why, From now on, companies and industries prefer rule-based chatbot systems. This document focuses on various topics related to the design, development, and implementation of rule-based chatbot systems. The main contributions of this document are the following:

- This paper presents a critical analysis of various existing rules-based chatbot systems.
- The paper discusses the issues of implementing chatbot systems and guides organizations in choosing the appropriate chatbot platform and framework
- The paper presents case studies on the two most popular rules-based chatbot systems, namely Google Dialog flow and IBM Watson. The document compares both frameworks based on several parameters. This document is structured as follows: Section 2 covers the historical development of chatbots. Section 3 describes and compares chatbots implementation platforms and development frameworks. section 4explains the parameters used to measure the performance of the chatbot system. Details of the design of Google Dialog flow and the IBM Watson Chatbot system and their limitations are discussed in Section 5. Section 6 contains the conclusion of this study and also discusses future expectations. Chatbot-System.

II. HISTORY OF CHATBOT SYSTEMS

All illustrations must be numbered. ELIZA was the first chatbot developed by Joseph Weizenbaum in 1966. ELIZA uses simple matching patterns. ELIZA is designed to work like a therapist asking questions. and reply to the user. Kenneth Colby founded PARRY in 1972. PARRY has the same reaction pattern structure as ELIZA but with an expanded control structure. It is used to simulate diseases. This chatbot system also has language understanding features. PARRY has effective variables such as anger, fear, and distrust. ALICE (Artificial Linguistic Internet Computing Entity) was founded in 1995 by Richard Wallace. It was inspired by ELIZA. It is an open source natural language processing chatbot program that interacts with a human by using some pattern matching rules. ALICE resides in XML knowledge base. It coordinates the client's contribution based on a predefined set of responses. Because you have a predefined set of QA reactions, you cannot successfully answer every query. ALICE's bots can respond. Extend your knowledge bases with XML. In this way, an ALICE bot can be a specialist in a specific data space. Google Dialog flow is a chatbot system from Google that was developed in2010. It is based on natural language conversations. These chatbots are able to respond to voice commands. Dialog flow is a Google service that runs on Google's cloud platform. The user can interact with the chatbot system via a voice and text-based interface. Watson is a Q&A framework developed by the exploration group at IBM's Deep Q&A company. IBM Watson According to Mind bowser's Scouting study for Chatbot Journal, IBM Watson is the most important decision-making phase for bots for 61% of companies. Microsoft Bot Framework Helps user to connect the chatbot to different channels like SMS, email and others. The Microsoft Bot Framework has its own software development kit (SDK). Popular Chatbot Platforms and FrameworksWhile implementing rule based Chatbot systems two key terminologies are Chat-bot development frameworks, and Chatbot platforms. Many times people get confused in these two terminologies.

Below table 1 shows differences between the frame-work and platform

Table 1 - Difference between Chatbot development frameworks and Chatbot platforms.

Chatbot Frameworks	Chatbot Platforms
It has a predefined set of functions and classes.	Chatbots can be installed and interacted with in an online environment.
Chatbot framework aids in the development and binding of Chatbot system components. It provides a tool that aids in the writing of code in a more efficient and timely manner.	Chatbot platforms provide as a foundation for deploying and running chatbot applications.
It requires programming languages to develop.	With the aid of a simple drag and drop technique, creating a Chatbot is straightforward and easy to grasp.
The Chatbot Framework allows you to store data and provide analytics.	The Chatbot publishing platform is the means through which users may access and use the Chatbot.
Developers and hackers use this programming language to create Chatbots.	Beginners and non-technical individuals can utilise it. Without coding, a user may create a Chatbot.
Example: Microsoft Bot Framework, Api.ai., Wit.ai, Google Dialog flow, IBM Watson, RASA NLU etc.	Example: Chattertypeople, Botsify, Chatfuel, Motion.ai, etc.

III. PERFORMANCE EVALUATION OF CHATBOT SYSTEMS

It is necessary to evaluate and compare the performance of a chatbot system. There are several dimensions that we can use to compare chatbot systems. Various returns are discussed in this section. Measurement parameters suggested by [16], [17] and [18]. According to [16], the following parameters must be taken into account to evaluate the performance of the chatbot system: Variety of tasks of the chatbot, number of channels (Facebook, Skype, Telegram, etc.)-compatible, compatibility with NLP and language, intelligent collection of user data through interaction, available tools for creating chatbots, etc. [17] proposes the following parameters to evaluate the performance of the chatbot system. Total number of users using a chatbot, number of messages exchanged per day, new users recently registered, how many times the chatbot gave a "don't know" response, user rating at the end

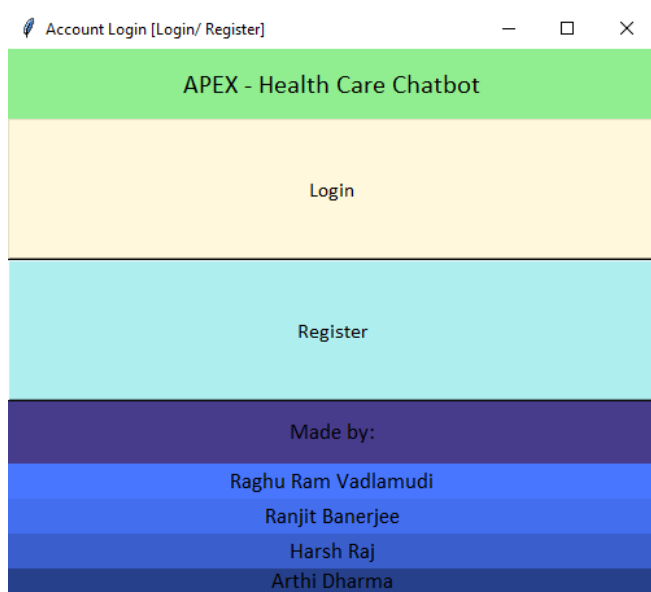
of the conversation, etc. [18] The following parameters are listed for performance measurement: system price, service credibility, usability, exit strategy when the chatbot cannot answer, etc. After summarizing the above parameters, we believe that the following parameters are most important when evaluating the performance of a chatbot system: multitasking, multi-channel support, support for NLP (Natural Language Programming), price, ease of use, Language/Applications/Integration, Free to Explore, Text-Based Query Support and Flexibility, Language-Based Query Support, Total Users, Engaged Users, Retention Rate, Backtracking Rate, Ease of Use, System Elasticity, and Scalability. Most of these features are supported by Google Dialog flow and the IBM Watson chatbot system. We present a detailed case study of these two chatbot systems in the next section.

IV. CASE STUDY ON THE IMPLEMENTED CHAT BOT

In this section, we will talk about the implemented chatbot built with Python and machine learning using CSV sheets where the data to the bot is trained to make the chatbot respond to the patient's health problem solving as described above can. Chat Bot Build is an offline chatbot that you can use to help patients when there is no internet. This chatbot acts as a primary healthcare helper where we can avoid doctor appointments due to minor illnesses or symptoms and consists of an internal doctor referral system that refers to the best-certified doctors when booking an appointment, thus reducing doctor search time. The chatbot created is a lightweight application that can be used anywhere. This chatbot can be easily updated from a new disease where people come from an unknown detection to a chatbot like Corona or a new disease. New disease records on entered existing diseases and a trained model to get an output

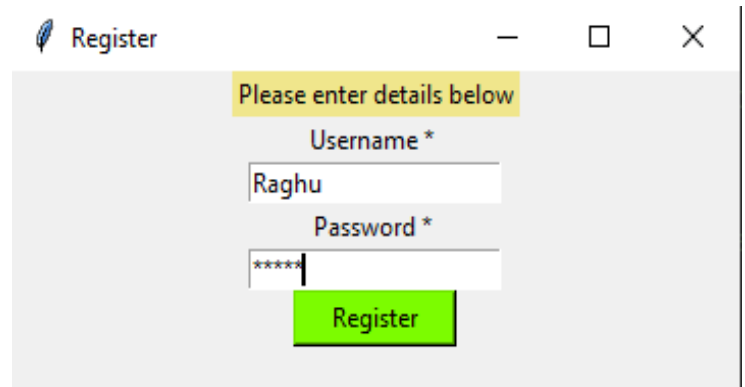
V. PROCESS OF THE CHAT BOT AND THE OUPUTS

The chatbot consists of the following steps:

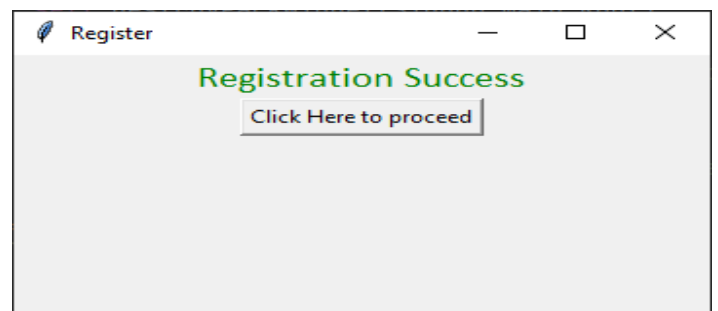


Fig[1]

Step 1: Fig[1] This is the main home page where the user needs to register to continue. The registered customer details are stored in the backend, which is used to verify the login details.

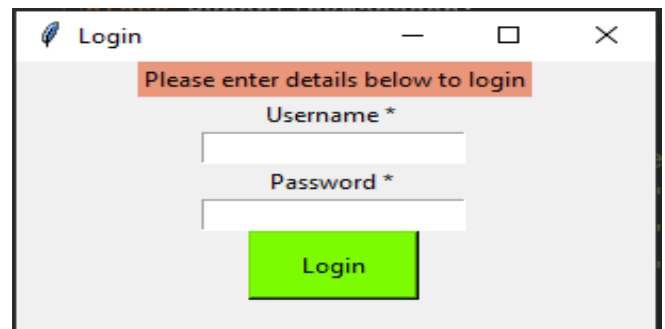


Fig[2]

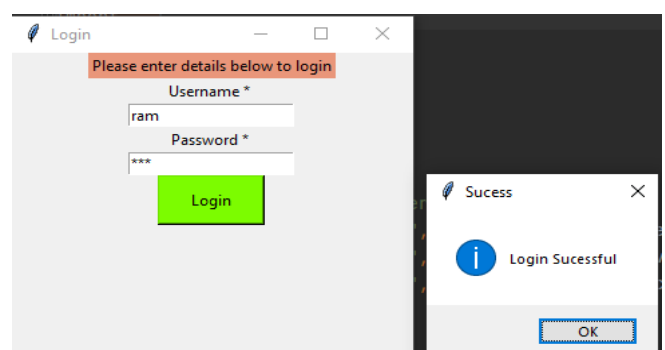


Fig[3]

Step 2: In this step, after registering a customer, a dialog box appears with the message "Registration successful" Fig[3] and a button "Click here to continue" Fig[3] that redirects to the main page of the chatbot



Fig[4]



Fig[5]

Question

slurred_speech ?

Question

Diagnosis

Please Click on Yes or No for the Above symptoms in Question

No Yes

Clear Start

Question

slurred_speech ?

Question

Dignosis

You may have :['Hypoglycemia']

symptoms present :['slurred_speech']

symptoms given: ['vomiting', 'fatigue', 'anxiety', 'sweating', 'headache', 'nausea', 'blurred_and_distorted_vision', 'excessive_hunger', 'drying_and_tingling_lips', 'slurred_speech', 'irritability', 'palpitations']

confidence level is: 0.06333333333333333

The model suggests:

Consult ['Dr. Jyoti Arora Monga']

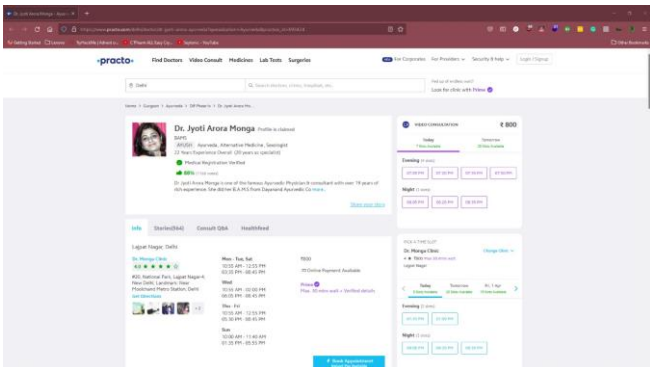
Visit https://www.practo.com/delhi/doctor/dr-jyoti-arora-ayurveda?specialization=ayurveda&practice_id=693434

No

Yes

Clear

Start



In this document, we first explain the chatbot system and its real-time applications. Here is some basic information about the chatbot's development frameworks and publishing platform. Several dimensions on which these platforms can be compared, such as features, programming languages , and many more. From all previous comparisons in this document, we conclude that each chatbot framework has its unique characteristics. We present an in-depth study and analysis of Google Dialog flow and IBM Watson. IBM Watson is currently more popular than the Dialog flow chatbot system. future expectations Chatbots are provided as follows. These expectations provide guidelines for researchers working in this field to improve the performance and behavior of the chatbot system. Currently, chatbots are not supported by

VII. REFERENCES

- ~ 4 ~