ISSN No:-2456-2165

Efficient Cloud Platform for Developing a Chatbot

Kunchala Little Flower[1], Veena Rani[2], Arun Kumar Kandru[3],
Professor[1,2], ,Department of CSE-AI&ML, Malla Reddy University,
Hyderabad, Telangana, India
Assistant Professor[3], Department of CSE, Malla Reddy Engineering College(A),
Hyderabad, Telangana, India

Abstract:- Intelligent conversational computer systems, such as chatbots, imitate human speech to provide online assistance and assistance in the form of automated responses. Many sectors have used chatbots to deliver virtual support to clients as a result of the enhanced advantages they provide. Natural Language Processing and Machine Learning are two areas of Artificial Intelligence that are heavily included into chatbots. Before the advent of chatbots, there were just bots. The emergence of chatbots has ushered in a new age of conversational computing. A method of conversion that is participatory. A textual chatbot is a computer programme that can have a meaningful conversation with a real person. A chatbot can be built on a variety of cloud-based platforms these days, including Microsoft Bot Framework and IBM Watson. However, each of these technologies has its own pros and cons (such as built-in artificial intelligence, natural language processing, conversion services and programming), so it's important to weigh the advantages and disadvantages of each before making a decision. Cloud-based chatbot systems are compared to determine which is the most effective, including those that include built-in artificial intelligence (AI). A comparison will tell which cloud platform is best suitable for constructing chatbots.

Keywords:- *Artificial Intelligence; Chatbot techniques; Cloud Platform; NLP;*

I. INTRODUCTION

From education to e-commerce, healthcare to entertainment, chatbots are being used in a wide variety of settings. If you're looking for advice in a variety of areas, chatbots may help you out [7]; this is the case with chatbots like Mitsuku, which are "small talk" centredchatbots that may develop a social relationship [8]. Chatbots, on the other hand, seem to be more enjoyable to the user than a static FAQ page on a website. Chatbots can assist many customers at once, making them more efficient and less expensive than human customer care representatives. Chatbots may be used to provide entertainment and companionship to users as well as customer support and assistance [9]. But the degree to which chatbots resemble humans in their embodiment and disclosure (how and when the nature of the chatbot is made known to the user) seems to affect users' interest in and trust in chatbots (see [10] for more information). [11]).

Both amusement and commercial uses may be achieved by creating the bots. We may classify the bots into Command-based bots and Smart bots depending on how we constructed them. A developer creates a command-based bot from scratch using input from the user. Because cognitive services aren't used to develop command-line bots, their capabilities are severely constrained. Artificial intelligence (AI) is at the heart of smart bots' ability to converse with people. According on the context and preceding message, clever bots are able to determine the best possible response.

Improvements in the implementation and assessment of chatbots are essential research subjects since they are widely used and implemented in so many different sectors. Deep Learning algorithms are a major emphasis of this paper's literature review, which is one of the paper's most significant contributions. The report also includes a proposal for future research in the field of chatbots, which identifies the obstacles and limits of chatbot installation and deployment.

Because of improvements in processing capacity and Natural Language Processing techniques and methodology, chatbots have evolved throughout time. In 1966, the creation of ELIZA resulted in the first successful deployment of a chatbot that mainly depended on language rules and pattern matching methods. Using a keyword-matching algorithm, it could connect with users. To get a response for the user, it looks for a good transformation rule to apply to the given input data. Eliza was a groundbreaking technology that sparked a wave of new study. ELIZA's scope of knowledge was limited since it depended on a limited set of context identification and pattern matching criteria.

Deep Learning algorithms have made it possible for chatbots to be used in a wider variety of contexts. As a brand-new application, the creation of intelligent personal assistants (such as Alexa from Amazon and Siri from Apple) is one of the most intriguing ones to date. Smart watches, specialised home speakers and monitors, and even automobiles now have personal assistants that can converse with the user through voice commands, such as chatbots or conversational agents. Users may wake up their smart personal assistants by saying certain words or phrases, and the gadget responds by turning on. Following an understanding of Natural Language Understanding (NLU), the assistant is able to respond to user requests, either by delivering information or by fulfilling activities, such as playing a morning playlist on Spotify (e.g., "Ok Google," "play my morning playlist on Spotify"). There are many tonal and geographical variances in human speech, as well as personal differences, that make interpreting human language difficult.

Cloud-based platform that may be used to build conversational AI systems is introduced in this study. Microsoft Azure bot service, IBM Watson, Chatfuel, Heroku, Kore, AWS Lambda, and more cloud platforms may be used to build chatbots. In the next part, I'll evaluate and contrast the features and functionalities of various cloud platforms.

ISSN No:-2456-2165

Azure Bot Service[2] provides an integrated environment[3] for deploying, creating, testing, and connecting to numerous channels that interact naturally wherever your users are engaged. IBM Watson Conversation[3] may be used to develop and train a conversational chatbot that uses intents and entities. Lambda functions in AWS Lambda[4] do not serve as a platform for Cloud Chat bots, but they make it easier to build and deploy them. An enterprise-grade PaaS, Kore[5] allows the development, creation, deployment and usage of superior, highly intelligent Natural Language Processing (NLP) bots across a wide range of different communication mediums. Chatfuel[6] is another service that makes it easy to create a chatbot without the need for programming knowledge. It's easy to connect to Facebook and other social networks using this one. Using Heroku[7], developers may create, operate, and test chatbots that are capable of using artificial intelligence and programming languages.[7] Heroku[7]

There are many different types of chatbots, each of which is meant to mimic human dialogue. This might be a written or spoken exchange. Most of the time, chatbots are used to collect data. However, it's more often than not accessed over the internet instead of a local network. Natural language input is used by the conversational agent to talk with individuals about a certain subject or topic. Software that mimics human behaviour is known as a chatbot. The creation of a response to the query is aided by a prior knowledge base.

Artificially intelligent software, or "chatbots," respond to user inquiries by posing new ones. In real-time, these enquiries might pertain to either students or employees. This project, on the other hand, promises to allow novices to create a cloud-based database programme and to provide students with answers to their questions. Do you want to know how this initiative is going to come up with the answers? When a student asks a question (input pattern), the chatbots will respond to that question using NLP i.e Natural Language Processing and ML i.e Machine Learning algorithms to provide an appropriate response.

II. OBJECTIVES

Chatbots made from scratch and constructed using a chatbot-building tool (Software as a Service) will be compared to see which solution is best suited for a gastronomy company employing cloud computing and software engineering practises. This research will evaluate whether or not a chatbot can be utilised to attract and retain customers for a gourmet firm. For this, a chatbot designed for one of the most popular messenger apps will be utilised in combination with current customer acquisition and engagement efforts. A chatbot should be able to provide information to a restaurant or coffee shop if it is asked for it. Its address, opening hours, and menu are all listed here.We finish with our findings and ideas for future study on the cloud platform based on some of the results.

III. METHODOLOGY

It's important to consider about how users and chatbots interact while developing chatbots. There are two things we need to consider: the user's message and the answer from the bot. The entire architecture of the chatbot is shown in Figure 2.The conversation starts with a message from the user indicating what the user is trying to say. Intent categorization and entity recognition will then be used to process the user input message. The user input message is checked by the intent classification module, which detects the message's purpose. The input message's context and the number of intents are used to determine the intents. The Entity Recognition Module of the Airline bot may be used to extract the city and date from the message structure and extract the core keyword from it.A bot's intent categorization and entity recognition modules are critical for determining the user's intentions and entities throughout the conversation. The answer generator is the next critical component in the chatbot design. The answer is generated by the usage of external APIs and algorithms. Intent and entities, as well as the conversation's context, are all gleaned from the most recent user message by the answer generator.

IV. ARTIFICIAL INTELLIGENCE CHATBOTS

In contrast to rule-based models, AI models use Machine Learning algorithms to learn from a corpus of human speech. As a result, a dataset for training the model must be created using Machine Learning techniques.Chatbots may now be more flexible and less reliant on specialised subject expertise thanks to the adoption of Machine Learning techniques,new pattern-matching rules may be automatically created and implemented without the requirement for operator intervention. As previously mentioned, there are two types of artificial intelligence models: information retrieval-based and generative.

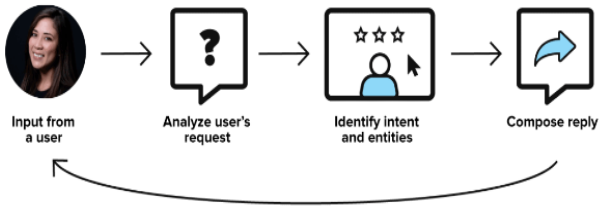

Fig. 1: How AI Chatbot works

Models of Information Retrieval The algorithms in Material Retrieval-based models are created to be able to retrieve the data a user inputs from a collection of textual information. Shallow learning methods are the most common, however Rule-based and Deep learning algorithms may also be employed in Information Retrieval models. Using an Information Retrieval paradigm, a chatbot analyses a user's question and selects one of the replies from a pre-defined collection of probable solutions. A database of question-and-answer pairs often serves as the basis for this model's knowledge base. Using this database, a chat index is created to provide a list of all the potential responses depending on the message that was sent. User queries are matched with information from the chat index using information retrieval techniques similar to those used for

internet enquiries. It is then given to the user in the form of a transcript of a chat. [16]. Because the replies are not created automatically, this strategy has the major benefit of guaranteeing the quality of the responses. With the rise of Web 2.0 and the availability of more textual data on social media platforms, forums, and chats, these models have become more popular.

For example, it may be expensive, time-consuming and laborious to build up the requisite knowledge base for this strategy. This implies that it will be more difficult for the system to match a user's input to the correct answer; it will need a significant amount of time and resources to teach the system to choose from a list of viable solutions.

Chatbots that use Information Retrieval systems, such as social chatbots, may not be the ideal option since they don't produce responses but rather extract answers from a pre-defined knowledge base. An key characteristic for this kind of chatbot [16] is that it is more difficult for Information Retrieval models to establish a personality. New information retrieval algorithms have recently been created, and it is vital to know which Machine Learning approaches are being used as the underlying technology for this sort of model. [31] proposed a unique approach for describing local textual co-occurrence and mapping hierarchical information across domains for more distant phrases in terms of semantic distance. To build this model, we assumed that the more often two phrases appear together, the closer they are likely to be connected across different domains. It is thus possible to use a high level of co-occurrence in a certain area to help in a search. Topic modelling for parallel text and obtaining a hierarchical architecture were therefore the foundations of this concept. The initial stage is to look for relevant patterns of word co-occurrence.

Proposes an innovative idea that tries to increase the quality and the veracity of the output by taking into account the preceding turn in the discussion, thereby acquiring additional contextual information. With the help of a Deep Neural Network, this model ranks not only the question/answer pairs that match the current user's input, but also those question/answer pairs that match reformed versions of previous debate turns.This model is based on After that, the rating lists for the various reformulations are combined. These prior queries may be utilised to get contextual information, which can then be used to acquire a more accurate response from the knowledge base[29]. Models that can generate new ones on their own. It's no surprise that generative models take the user's input and utilise it to build new replies, word by word. Consequently, these models are able to create new phrases in response to user queries; nevertheless, they must be taught to grasp sentence structure and syntax so that the outputs may be uneven in quality.

Standard Algorithms in the Industry, equence to Sequence models for chatbot modelling are now the industry standard among AI models. They were first developed to address issues with machine translation, but it seems that the core ideas work just as well for NLG. The encoder and decoder in these models are both Recurrent Neural Networks (RNNs). It is a hidden state of the RNN that processes each word of the user's sentence. It's the conclusion of the sequence, and it tells us what we're going to do next. One word at a time, the Decoder builds a new sequence (or phrase). A objective of this probabilistic model is to utilise the preceding turn's input as conversational context, and then learn how to offer a response that is as near as feasible.The response or output phrase is delivered to the model in the learning phase so that it may learn through back propagation. The learning phase. The interference phase may be approached in two distinct ways. The output sentence is chosen using the greatest probability among the input sentences provided by the beam search technique. The predicted output token is used as an input to predict the following phrase in a greedier method. This strategy employs the expected output token.

V. CHATBOT PROTOTYPES

The Messenger Platform Core collection of bot-building APIs was selected to meet the research's goals. MongoDB and IBM Cloud (Platform-as-A-Service) were used to construct the chatbot's back-end server for handling request allocation, which was deployed as cloud-native application. The cognitive capabilities of the chatbot were examined for further development. The chatbot was constantly improved until it met all of the customer's needs. In order to better understand the user's intentions, a simple NLP engine was implemented (i.e., buttons, a picture gallery). Figure 2 depicts the system's architecture for sending messages over the Messenger platform by a user (mobile or browser-based). In order to handle the public data of the user, the Facebook page is connected to a webhook. Even if the user isn't in the database, the information on him or her is kept.

There are now a number of cloud-based systems that allow us to build and deploy our own bots. There are cloud-platforms that provide other services than bots, such as cognitive services and built-in artificial intelligence. Our chatbot was built on the Microsoft Azure cloud platform, as detailed in this article. There are three components to Microsoft's bot framework: A directory of bots, a bot builder, and a bot connecter. If we like, we can even run it through its paces using the supplied emulator. It comes with a Node.js and Net bot building SDK. LUIS (Language Understanding Intelligence Service) [14] may be used to enhance the interactivity of our bot. Using the messaging applications, users may connect with the bot using Microsoft's Bot Framework, shown in Figure 3.[15, 16, 17, 18, 19] The significance of the bot connector in Microsoft's bot architecture is critical. Facebook, Skype, and other social media platforms are among the channels that the bot connector connects to.

ISSN No:-2456-2165

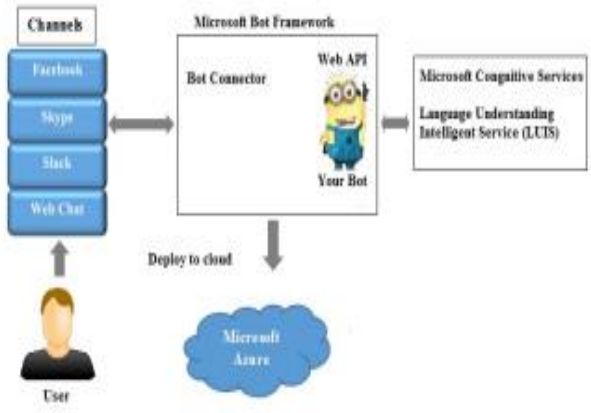

Fig. 2: Microsoft Bot Framework

VI. RESULT AND ANALYSIS

A supported cloud platform for building a chatbot has previously been mentioned in the first part. Chatbot cloud platforms come with a slew of benefits and drawbacks depending on how they're used. All bots developed in various cloud settings and their outcomes are compared in this section. In this article, we'll focus on the most popular ones, including Microsoft Azure bot service, IBM Watson, and Heroku among others. To better understand how the chatbot is built, we present the integrated environment and the analysis table that are utilised for this purpose.

Channels	Programming Languages	Artificial Intelligenc	Integrates with	Pro's	Con's
Slack Facebook Messenger Skype GroupMe Telegram Twilio	Node.Js,C#	Built in - LUIS	Any API via programming	- No IDE required Integrated environment - Runs on Azure serverless architecture (Azure Functions) - Well documented	- Runs currently in preview mode - Required Azure subscription

Table 1: Analysis table for Microsoft Azure cloud platform

Channels	Programming Languages	Artificial Intelligenc	Integrates with	Pro's	Con's
Slack Skype Web Mobile Email SMS	Node Ruby Java PHP Python Go Scala Clojure	None	SFDC as its part of Heroku ecosyste Any API via programming	- Secure (https) - Endpoint,Integrated with GIT so simple deployment., - Single platform for development & deployment	- Complex - Steep learning curve - Programming skills required.

Table 2: Analysis Table for Heroku Platform

Channels	Programming Languages	Artificial Intelligenc	Integrates with	Pro's	Con's
Facebook WeChat Telegram Kik Line Kakao	NodeJS Python Java Unity Android IOS	Built in Watson Conversation Other Watson Cognitive Services	Any API via programming	- Minimal number programming required - High Quality of Interaction - Proven solution	- Limited option to integrate various Watson services - Static Dialog

Table 3: Analysis table for IBM Watson

Attributable to the nature of the system (Messenger Black box testing was performed using a logic-based method (causeeffect), where each user intent results in a specific chatbot action being performed. In both situations, all of the functional requirements were put to the test by including the intended audience in the system and acceptability testing processes. Tests of the chatbot prototypes (PaaS and SaaS implementation) were done with university canteen customers who were selected at random. Prototypes' weaknesses were discovered via both direct observation and customer input. Iterative improvements were made to the PaaS prototype after each test (iteration) and new users were recruited as part of this effort to better serve its customers. During the most recent iteration, we made use of the SaaS prototype.

It was possible for the chatbot to send messages to all users who were available on random days and ask them to nicely answer a few questions. Based on the information provided by the users, segmentation was carried out. Table 1 displays the final number of users and the primary segments. More segments were discovered by using a chatbot to exclusively pose questions to students and receiving just 71 responses. Only four students said they'd used chatbots before; 59 said they'd use one to make an order, 3 said they'd pay with one, and 65 said they'd use one to obtain a free drink. According to a survey of users, WhatsApp is the most popular messaging app, with 69 respondents, followed by Telegram with 20 respondents. Despite the fact that none of the users (53 of them) designated Messenger as their primary messaging programme, many nevertheless use it on a regular basis (36 users). User input was the primary source of qualitative data that was analysed and used to enhance the chatbot prototype's usability and user experience. While some data was gathered at Mensa, others were supplied by users in the form of SMS messages by way of the chatbot's Feedback feature.

VII. IMPLEMENTATION OF CHATBOTS

The PaaS prototype was refined until it met all of the functional specifications and was put to the test with real-world users. Because the NLP engine does not follow sophisticated AI algorithms or have ML capabilities, it was unable to execute adequately in several cases. The NLP engine was unable to interpret the user intentions on several instances when messages were written in German and the NLP engine was solely tuned for English. More usable for users was found to be an image gallery-style chatbot menu with choices shown as buttons. When tested in accordance with ISO/IEC 9126-1, this PaaS prototype was found to meet all of the requirements for usability, reliability, efficiency, system architecture compliance, and deployment on any PaaS supporting Node.js applications and NoSQL data. As a result, users were able to make use of the chatbot and receive accurate information when doing so (functionality) (portability)

All functional criteria were met in a short amount of time owing to the drag-and-drop functionalities used to build the SaaS prototypes. Some of the non-functional tasks, such as hosting the chatbot, creating a webhook, and setting up HTTPS ports and a database, are taken care of by the chatbot building tool. AI engine was used to combine NLP capabilities, however certain user inputs couldn't be processed and German messages couldn't be processed since the tool only knows English. Some users were able to input text while others were unable to do so in Messenger, which was operating on iOS, Android, and a few browsers (Safari,

Chrome). Purchase of a PRO version is recommended by the tool's instructions to fix the problem. Despite the fact that the writing tool makes it simple to create chatbots, neither the instructions nor the tool itself are very versatile. In order to send messages just to people who are celebrating their birthdays, special code would have to be executed, which the tool does not provide. Because of this, it is not feasible for chatbots to export or import user data from other chatbots. Which strategy is more suited to a restaurant? It all depends on the situation. Many factors need to be taken into account, including the number of initial users, who will manage the installation and maintenance, and how quickly the system has to be operational. The primary objective of the chatbot and the sort of information it provides to clients are crucial considerations to make when deciding on a strategy. In the Mensa, for example, the menu is changed every week, which means the photos, descriptions, ingredients, pricing, and nutritional information for the various dishes are all subject to change. A PaaS solution is more suited for this sort of eatery. In addition, PaaS deployment would be great if there will be active advertising and interaction via alerts and segmentation. Small restaurants and cafeterias with consistent menu items and prices throughout the year would benefit from a chatbot authoring tool, even if it was free to begin with. The number of users and the benefits they stand to gain from an upgrade would determine whether or not an upgrade would be worthwhile.

VIII. CONCLUSION AND FUTURE WORK

Software has developed and will continue to evolve in much the same way that hardware, from computers to mobile and wearable devices, has. Mobile user interfaces have grown cleaner, simpler, and easier to use, and chatbots are becoming more widespread as a means of interacting with software. No matter how you feel about artificial intelligence and machine learning, these technologies will only improve in the next years. When it comes to social media, companies are already integrating chatbots into their Facebook business pages as well as their websites. When it comes to providing customer support, chatbots will soon be a need. In this study, it was shown that people are eager to experiment with new technology and utilisechatbots if they see benefit in doing so. Similarly to how restaurants employ online and mobile apps to deliver services, chatbots will become another channel for communication with the potential advantage of reduced deployment and maintenance costs than web and mobile applications. The use of containers to run distinct system modules makes it simple to manage and protect applications regardless of the underlying infrastructure. If one module fails, other modules may continue to operate while the failed module is restarted. This strategy is more efficient than the alternative. Dynamic workloads and pay-per-use computation time may be achieved in milliseconds using cloud-native features such as Serverless. For future work, the chatbot that was built using cloud software engineering principles is being rebuilt so that it may be used on other instant messaging platforms.

REFERENCES

[1.] Kehoe, Ben, SachinPatil, Pieter Abbeel, and Ken Goldberg, “A Survey of Research on Cloud Robotics and Automation”, IEEE Transactions on Automation Science and Engineering, Vol.12, No.2, (2015), pp.398- 409.
[2.] Creating your first bot-Documentation-Bot Framework. [Online]. Available: http://www.garypretty.co.uk/2016/07/14/ creating-your-first-bot/
[3.] Conversation-IBM Watson Developer Cloud. [Online].Available:https://www.ibm.com/watson/developercloud/ conversation.html
[4.] Create and Deploy a Chat Bot to AWS Lambda in Five Minutes AWS Compute Blog. [Online].Available: https://aws.amazon. com/blogs/compute/
[5.] Chatbot Intelligence-Kore. [Online]. Available: https://kore.com/ bots-platform/chatbot-intelligence/
[6.] Create Chatbot for free. Best chat bot platform Chat Fuel. [Online].Available: https://chatfuel.com/platforms
[7.] Cloud Application Platform-Heroku. [Online]. Available: https:// www.heroku.com/
[8.] Developer’s Introduction To Chatbots-Tutorialzine. [Online].Available:http://tutorialzine.com/2016/11/ introduction-to-chatbots/
[9.] S. A. Abdul-kader and J. Woods, “Survey on Chatbot Design Techniques in Speech Conversation Systems”, International Journal of Advanced Computer Science and Applications, Vol.6, No.7, (2015), pp.72-80.
[10.] S. A. Abdul-kader and J. Woods, “A Survey on Web based Conversational BOT design”, Journal of Emerging Technologies and Innovative Researchvol, Vol.3, No.10, (2016), pp.96-99.
[11.] M. J. Pereira and L. Coheur, “Just . Chat - a platform for processing information to be used in chatbots”,.[Online].Available:http://www. chatbots.org/ai_zone/viewthread/492/
[12.] V. A. Gandhi, C K Kumbharana, “Comparative study of Amazon EC2 and Microsoft Azure cloud architecture”, International Journal of Advanced Networking Applications, ISSN 0975-0290, (2014), pp.117– 123.
[13.] Chatbot Architecture. [Online]. Available: https://hackernoon. com/chatbot-architecture-496f5bf820ed#.mtmd52u8i
[14.] Bot Framework Overview-Documentation-Bot Framework. [Online].Available:https://docs.botframework.com/en-us/
[15.] S. Kumari, M. Karuppiah, X. Li, F. Wu, A.K. Das and V. Odelu, “An enhanced and secure trust-extended authentication mechanism for vehicular ad-hoc networks”, Security and Communication Networks, Vol.9, No.17, (2016), pp.4255–4271.
[16.] M. Karuppiah, S. Kumari, X. Li, F. Wu, A.K. Das, M. K. Khan, R. Saravanan and S. Basu, “A dynamic id-based generic framework for anonymous authentication scheme for roaming service in global mobility networks”, Wireless Personal Communications, Vol.93, No.2, (2016), pp.383–407.

[17.] M. Karuppiah, "Remote user authentication scheme using smart card: a review", International Journal of Internet Protocol Technology, Vol.9, No.2–3, (2016), pp.107–120.
[18.] M. Karuppiah, S. Kumari, A.K. Das, X. Li, F. Wu and S. Basu, "A secure lightweight authentication scheme with user anonymity for roaming service in ubiquitous networks", Security and Communication Net-works, Vol.9, No.17, (2016), pp.4192–4209.
[19.] X. Li, J. Niu, M. Karuppiah, S. Kumari and F. Wu, "Secure and Efficient Two-Factor User Authentica-tion Scheme with User Anonymity for Network Based E-Health Care Applications", Journal of medical systems, Vol.40, No.12, (2016), pp.268.