

Developing Chatbot Wrapper for Online Shopping: A Case Study of Using Generic Mobile Messaging System

Saketh Vaddadi ¹, Sahand Asri ¹, Yashar Ghemi ¹, and rajesh aytha ¹

¹Affiliation not available

October 30, 2023

Abstract

Developing Chatbot Wrapper for Online Shopping: A Case Study of Using Generic Mobile Messaging System. A conversational UI chatbot using Telegram API was implemented.

Developing Chatbot Wrapper for Online Shopping: A Case Study of Using Generic Mobile Messaging System

Sahand Asri
Department of Computer Science
Lakehead University
Thunder Bay, Canada
sasri@lakeheadu.ca

Yashar Ghemi
Department of Computer Science
Lakehead University
Thunder Bay, Canada
yghemi@lakeheadu.ca

Saketh Vaddadi
Department of Computer Science
Lakehead University
Thunder Bay, Canada
svaddadi@lakeheadu.ca

Rajesh Aytha
Department of Computer Science
Lakehead University
Thunder Bay, Canada
raytha@lakeheadu.ca

Sabah Mohammed
COMP5112WC Supervisor
Lakehead University
Thunder Bay, Canada
mohammed@lakeheadu.ca

Abstract—Chatbots started playing a key role in the present world market which highly rely on the internet, and social media on high demand, would embrace such technology. These chatbots or Conversational User Interface (CUI) help customers find useful information that they need in a reliable and efficient way. They are revolutionizing the modern era as customer service with 24/7 services and many organizations have already started using this technology. However, developing a bot that satisfies the humans' needs is challenging. In this paper, we discuss the bots we envisioned for online shopping and various available approaches. Then, we propose a system in achieving the goals for the chatbot to work as an assistant for online shopping. Finally, we deploy our rule-based chatbot on Telegram Instant Messaging Application (IM) as developing a chatbot from the scratch, we need to implement our own messaging application which is challenging and not popular as current IMs.

Keywords—Chatbot, Rule-based, Shopping Assistant, E-commerce, Instant Messaging Application

I. INTRODUCTION

The development of technology, especially the internet, allowed people to interact quickly with Instant Messaging (IM) applications such as WhatsApp, Facebook Messenger, WeChat, and Telegram. These applications are designed to provide an environment for people to deliver their message. Also, some of them present communication channels to broadcast and spread the information for the users.

In addition, people are using the internet to arrange some of their daily tasks at home, such as online shopping. As a result of growth in the usage of instant messaging software, nowadays, people prefer to chat to explain their needs and expectations instead of physical communications. Besides, most of the companies do not have enough human resources to respond to the users' demands. Therefore, chatbots were offered to deal with this issue.

A chatbot, as the name itself suggests, is a combination of chat and robot. In simple terms, a chatbot is a robot that has the capability of conversing with a user either through text or voice. Many applications like Skype, Spotify and Wechat etc. use chatbots, which has a huge market. It is estimated that by the end of 2020, most of the world market will start implementing chatbots for their businesses. The main benefit of an automated assistant is that it is available all the time to help the customer as a virtual customer service agent. Alongside, a chatbot is mainly seen as a customer service

assistant; it can also be used within the organization to help employees work easy and also accelerate production rate. The idea was first introduced in the 1960s by MIT professor Joseph Weizenbaum which was called ELIZA [1]. But in later years, rapid growth in text and natural language interface lead to further development in chatbots.

After the rise of the internet the demand for online shopping has surged and even chatbots are starting to gain their importance in the field of E-commerce. For the last decade, we have seen the demand for online products and could see companies like Amazon, Costco, eBay and many more rise to billion-dollar companies. But we need a huge number of resources at the backend to run the organizations efficiently to satisfy customer needs. Secondly, to resolve the issue faced by the customer might take time, let's say if a customer had an issue and the only option the customer got is to contact the service agent, try to convince them and so on which is a time taking process but if a conversation AI bot is introduced into the system it would have verified and solved the issue in no time. Not only can bot help in customer service, but also it helps customers find the right product they need. Similarly, since dealers and their consumers are not required to meet each other, their interactions are normally completed by exchanging electronic mail or via utilizing an electronic messenger. The client may additionally ask numerous questions before deciding to shop for a product. If the seller actively interacts with the client, it may well increase the popularity of the vendor whilst the consumer has a tendency to purchase from a reputable dealer. The purchaser might also ask some questions earlier than deciding to buy a product. If the vendor actively interacts with the patron, it is able to boom the recognition of the vendor while the client tends to buy from a good seller.

Thus, the seller responsiveness in answering any customer questions plays a vital role in the success of a transaction. Nevertheless, in numerous conditions, let's say in holiday or at some stage in a rest, the seller can also not be able to talk with the consumer. As a solution, the vendor can employ several purchaser services to answer any questions from the patron. However, it may entail additional fees for paying customer support officers. As an alternative solution, we can hire a gadget that may get hold of and apprehend the query from the client and, automatically, giving the exceptional response in accordance to that query.

Fortunately, in the last decade, many researchers are actively involved within the area of artificial sensible and its related research topic including system getting to know and natural language processing (NLP). With the one's advancement, we may be able to develop a talking robot or chatbot in short, that's an automatic program that can interact with its spoken counterpart with non-human intervention. In order to build one of these programs, there are numerous challenges stand up, including how to recognize the requested question even though it includes any mistyped word, and then pick out the excellent among several viable responses. Now to achieve good accuracy and, at the same time, efficiently responding to the users either through text or speech is a big challenge.

Generally, there are two types of chatbots based on their responses, which are generative-based chatbots and rule-based chatbots. Generative-based chatbots are not using predefined answers to respond to the users. Instead, they are making use of deep neural networks that are trained by a large number of previous conversations. These types of chatbots are fit for conversations. However, they are not suitable for goal-oriented tasks, where users are looking to receive a specific and precise answer since the responses of generative chatbots could be inappropriate and unexpected. In addition, the training cost of the deep neural network model is high and demands a huge number of conversations, which in some cases could be impossible to collect.

Unlike generative-based chatbots, rule-based chatbots use predefined repositories to respond back to the user. Rules-based chatbots should have the ability to classify or select the best answer out of the responses in the repository, based on the users' questions and demands. Rule-based chatbots are suited for goal-oriented tasks since they could assist the user in finding the proper answers using human-computer interaction (HCI) methods. Furthermore, they hardly make syntactic or semantic mistakes.

Chatbots are also divided into two categories based on their domain: open-domain chatbots and closed-domain chatbots. Open domain chatbots are the ones that operate and optimistically understand every topic and words in a language. On the other hand, closed domain chatbots are only operational in a particular topic and language. Open domain chatbots are fundamentally using a trained model to generate sequence to sequence (seq2seq) answers for the end-user. In other words, building a rule-based open domain chatbot is practically infeasible. In contrast to open-domain chatbots, closed-domain chatbots do not necessarily need a learning model to choose or generate answers.

In this project, we are implementing a rule-based shopping assistant chatbot named ShopBot which works on a closed-domain. One way to develop such shopping assistant chatbot is to build it from the scratch without the use of available messaging applications and chatbot frameworks. This is quite challenging and unprofitable for several reasons. First, to develop a chatbot from the first beginning we must also develop an instant messaging application and handle all the users' conversations. Instead we can use the chatbot frameworks which were designed by available messaging applications to build our idea which is a smarter choice. Second, even if we achieve the first problem, our messaging application is not yet popular enough for users. Therefore, probably the number of people using our chatbot would be

limited. In addition, we could assert that the users who do not prefer to install a not frequently used new applications. The next problem is with the sign up process. Most of people are unlikely to share their personal information with different vendors. To extend the sign-up problem, another chore of users is login task, memorizing the different login. Additionally, despite the execution environment every new application serves new user interface that major portion of its user need to learn it. With the consideration of all the given reasons, this chatbot has been developed by using the Telegram bot framework which can provide reliable communication between our users and our server application.

II. RELATED WORKS

Shopping by itself is an interesting task for many, not only because they own a new product also because its task, interaction and conversation with the shopping associates, probing through different qualities and product and so forth. These qualities rise the interest for the buyers to go through shopping chatbots via using natural languages aided by commands, buttons, Images and others visualization materials. According to the aforementioned, shopping assistant chatbot can absorb a plenty of naïve users who are only familiar with the messenger or those are not comfortable with online shopping applications. Moreover, the interactional and conversational environment are more likely to many in compare to the solid web pages or mobile applications. According to [2] most of the customer questions in an E-commerce website were related to business, coupons and logistics followed by the other needs. With the development of deep learning and reinforcement learning, recent researchers have shown interest in developing chatbots based on these new approaches but the problems lie within as the it is for the industrial application with huge data, being relying on labeled data might not be efficient. The other problem is the users or customers expect the bot to be human like and respond to their queries just like a human being which in real time world is quite difficult to achieve. Instead, several research papers [3] were published regarding direct extraction of information from a pre-existing knowledge base. Web information extraction (WIE) using attribute entity paired wrapper methodology is one of the few examples. All the given reasoning brought us to develop a conversation interacting shopping assistant chatbot which extract taste, personal information, interests and requirements of the users, record the history of each individuals and understand market needs and demand to provide them products, offers and promotions appropriately.

Mhatre et al [3] concentrated on web-based bots where the conversation occurs through e-mail by evaluating through AIML files which requires the help of AIML and language processing algorithms such as pattern matching used in machine learning. Using Donna Interactive Bot, experiments were conducted around 700 messages, the timing response is between 7.5 seconds to 48 seconds with a bad response of 5%, an excellent response of 35%, and a response that is almost similar to requests of 60%. Overall efficiency of 70% is achieved in terms of system responding to question within specific time period which is very less because, when the input increases most of the system efficiency decreases which is a limitation for the system in responding has taken a long time to respond to client requests.

On the other hand, Pradana et al [4] used AIML and ELIZA in SamBot by implementing a log-based knowledge maintenance module measured using the Turing Test or Loebner Prize chatbot throughout the world by getting 132 points and a score of 26.4 points. In this research, 100 questions from online users were collected, but 30 questions that were inappropriate in responding. Based on this research, we came to know that information is only limited to botmaster knowledge so that the chatbot will respond incorrectly to the input and generate random answers when the questions asked are beyond botmaster knowledge that can interfere with users in interacting with a chatbot.

Md. Shahriare Satu and Md. Hasnat Parvez [5] showed a review of integrated applications with AIML based chatbot and research was conducted on the applications that using AIML chatbot instead of human representative to interact with customers and giving answers to the specified problem. Also discussed about the cost factor, implementation and configuration of AIML based chatbots.

Siddharth Guptha [6] used Rivescript, a scripting language to fetch information based on user input. A Rivescript is a plain text, line-based scripting language with an intent to learn, type and read easily. The site has conventional navigation choices for the client on the off chance that the individual in question wants to peruse the site, in a conventional way. It will likewise highlight conventional search/filter alternatives. The propose website will have an unpretentious chatbot that can be gotten to from any page. The client can collaborate with the bot utilizing Natural Linguistics. The bot can make recommendations, give data, or pose further inquiries to evoke additional data relying upon the client cooperation. However, the bot has an acute response time. Here the server contains a PHP file that serves as a mode of interaction between the Chat Client and the MySQL database. At the point when a trigger that contains an object macro is called, the response is parsed and afterward executed by the Interpreter's JavaScript Object handler.

Lui Lin [2] developed a chatbot called WebChat which was written in python under tornado framework. The communication is between client and server is in JSON format and users can access the website in their browser through web sockets.

None of the mentioned works tried to implement their chatbot for a popular instant messaging application and their APIs to develop their chatbot. Therefore, fewer number of users will use these chatbots. Instead, we plan to implement our chatbot using Telegram, which is one of the most popular instant messaging applications in many regions of the world and for that, we use Telegram Bot API.

Furthermore, the mentioned works are basically learning and generative chatbots. For conversational chatbots, even high error rates and responses are acceptable. However, as we explained earlier a Shopping Assistant Chatbot needs to be goal oriented to meet the users' desires. Therefore, designing and developing a Rule-based Chatbot for this task is more effective and more efficient. For that, our proposed chatbot is a Rule-based Chatbot while benefiting from some features of the conversational bots.

III. PROPOSED METHOD

The design of this application took placed on different scopes to accommodate the users' needs. The first and main module of the system is request handing which can be in form text, commands and selection between choices. The other majority is the way a conversation and its respective flow is controlled; in addition, the history of requests is being kept. The final module related to analyzing and using the searched requests for future responses.

Despite the aforementioned, another critical concept is the platform via which our users could perform the commutation with ChatBot engine.

A. Telegram

Telegram is one of the popular messaging applications working on different operating systems and platforms namely: Android, IOS, windows, Linux and web browsers. Also, it is notable that Telegram is an open source program initially developed for mobile devices. However, many other features and APIs have been added. One of which was Telegram Bot API. There are 3 parties of Telegram involving in development 1) Telegram BotFather 2) Telegram Bot API 3) Telegram Bot.

1) *Telegram BotFather*: BotFather is a Telegram bot has been developed by Telegram team itself to provide facilities to create a new Bot or manage existing one. Additionally, it issues certificates and token for the created token.r

2) *Telegram Bot API*: It is set of function in different languages, almost most of the well known programming languages, to recive and sending messages from users. The format of messages that can be handled are varing, majorly named: text, voice, images, commands and so forth[7].

3) *Telegram Bot*: It is perse a machine acting like a user. It can recive messages, process them and provide the aproprate answer regarding the incomming message. The general architecture of Bot has been constituted by a number of different states under which several tasks and new states is presented.

B. Chat States

The terminology of state in Teleram Bots refers to the latested predefined command of syntax that requested by the user. In each particular state only the designated tasks to that state would be executed. The main conception behind the states is to provide a tree of commands and buttons to guiding the users correctly to what they want to get. The figure.1 shows the predefined states of the chatbot which is same for each users.

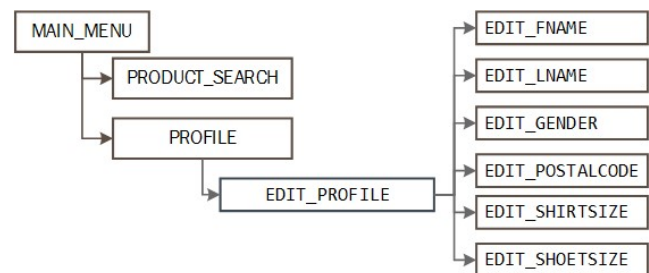


Fig. 1. ShopBot Message Handler States

C. Messaging and message types

The main element of any chatting system is message, it is therefore at the highest level of importance. The message itself can be studied from three different perspectives which are the sender, the intention, and the content and format.

By consideration of sender end, there are two parties who are involving in process of exchanging messages. One end is the ChatBot engine. However, it is a machine, it is capable of sending material like a user. The Telegram Bot APIs provided more sophisticated materials for Bots in compare with the human users. As an exemplification we could name card, font styles and buttons. The users are suppose to send three types of messages to chatbot engine those are text, predefined buttons and link buttons.

D. Conversation flow and History control

Conversation flow is commonly referred to the present message exchange that took place. When a conversation starts for particular product the chatbot tries to extract the user needs and intention. By this way however the features and properties may change the product name is still same. As soon as the user ask for different product chatbot start new conversation.

Telegram messenger stores each conversation with chat_id and user_id. Additionally, each message is represented with message_id. With the given information we could recover all the messages at any time. But many of those messages might be useless for further use. Another important module of this system is called Business Intelligent Engine. The BI engine role is to analyze the users request and finding facts related to market demand as well as every single user interest.

E. Message Handler

One of the main components of ShopBot is the Message Handler module. This module is responsible for message capturing, message categorization, and state transition. Message Handler module has two methods to deal with users' messages. In the first method, the module benefits from regular expression patterns. It should be noted that the regular expression patterns depend on the current state of the chatbot. For instance, suppose the chatbot is in the MAIN_MENU state. As mentioned in Fig. 1, the chatbot may transit to PRODUCT_SEARCH or PROFILE for as the next state. Therefore, it expects words like 'product', 'search' or 'profile'. As shown in Fig. 2, if the user's message contains one of the first two words, it will call a search_product method. Then, this method performs its tasks, and after that, the chatbot transit to PRODUCT_SEARCH state.

```
state.MAIN_MENU:
[MessageHandler (Filters.regex('^.*(Profile|profile).*$'),
  profile.profile),
MessageHandler (Filters.regex('^.*(Search|search|Product|product).*$'),
  search_product),
MessageHandler (Filters.regex('^.*(About|about)$'),
  about)],
state.PROFILE:
[MessageHandler (Filters.regex('^.*(Display|display|Show|show).*$'),
  profile.display),
MessageHandler (Filters.regex('^.*(Edit|edit|Modify|modify).*$'),
  profile.edit),
MessageHandler (Filters.regex('^.*(Back|back|Main|main)$'),
  start)],
```

Fig. 2. MessageHandler for Main Menu and Profile

F. Product Search

One of the most important part of any shopping assistant chatbot is the product search since the results that a user expects to see has to be the extracted by this module. [8]. As

Amazon is a popular and well-respected company for online shopping, for this task we are using Amazon Web Services (AWS). One of the necessary operations of AWS for this project is the ItemSearch operation. ItemSearch searches for products on Amazon and in response Amazon API returns up to ten items per each page.

The chatbot expect the user to write down a query about her/his demand when approaching the product search module. Then, it reads the user request and collect the useful data using the predefined shopping dictionaries we designed for the bot. Then, the chatbot try to understand the user's need by gathered information and it will send a request to ItemSearch operation of the Amazon API and wait for the response. After receiving the response from Amazon API, the chatbot will extract the product items and their properties. Then, the product properties will be compared to the stored profile information and the chatbot will display the best options to the user based on the user's profile information and history.

IV. PROTOTYPE

As a result of increase in the number of instant messenger users, we propose a prototype of ShopBot to aid the costumers to choose the right product. The main characteristic that this project adhered to develop a wrapper for Telegram that uses a chatbot engine to control and make communications with users. The other functionality needs to be accommodated in this project is product search and visualization of product in the wrapper.

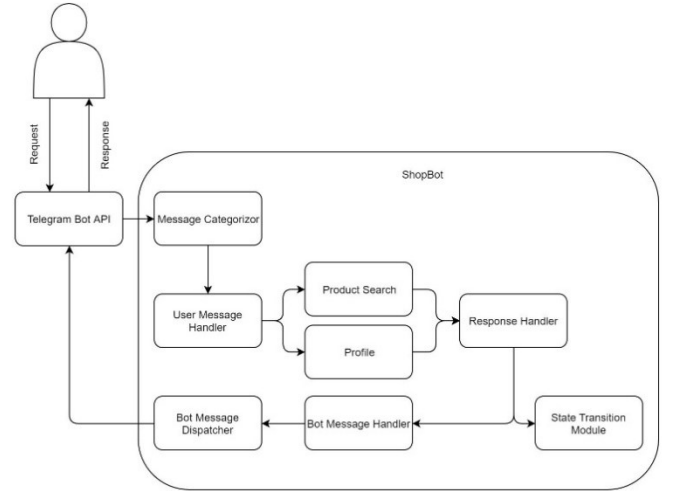


Fig. 3. ShopBot Flow Diagram for Product Search

Fig. 3. demonstrates the Product Search Flow Diagram of ShopBot. In the beginning of the flow, the user requests some change or information using Telegram Bot API. Then, Telegram Bot API handles the message distribution and sends the request to the ShopBot engine. First, ShopBot Message Categorizor tries to classify the users' messages depends on the message and current state. Then, the User Message Handler manage the input messages by sending them to different modules which are product search and profile. After that, the response handler, uses the profile information and product search results to find the best responses (selected product results). Then, the State Transition Module changes the chatbot state depends on the message and current state. At the same time, the Bot Message Handler creates a Bot Message using the collected responses. Finally, the Bot

Message Dispatcher transmits the Bot Message to Telegram Bot API which will be send back to the end user.

A. Messages:

As to the nature of chat systems the messages are element that they exchange between the end users. In this particular case we have mainly two different types of messages which are End User Messages and Bot Messages.

1) *End User Messages*: These messages are could be commands or scripts. To start the conversation, the rule is that user has to type command “/start”. Similarly, to terminate the current session user must type “/done” or “done” command. These two commands have been depicted in Fig. 4.



Fig. 4. Start Command and Buttons

As soon as the “/start” command performs the telegram bot receives the message that is containing the information about the user and other additional information. If the user’s information was not available in the chatbot, then the existing user’s information which was previously stored in Telegram (i.e. user_id, name) would be added to profile data set. All the profile information can be edited by the user anytime in future through profile section.

Despite the given instructions the other predefined commands are available in form of buttons and accessible through tree of commands. However, the system is capable of receiving script from users and by applying regular expression on them and map them to most appropriate commands.

Buttons mainly perform a task. The first one is to change the state one step forward or backward. The second is to perform a command at present state. In Fig. 4. the profile and search product buttons move the conversation to the next state as it is showing in Fig. 1. On the contrary “About” button only perform the task of showing information about the current state of the chatbot. Action of this command would not change the state and it remains in the MAIN_MENU state.

The other form of messages is scripts or texts. One part of these messages are being handled by regular expressions to be mapped to a particular command. The other part will be handled in Search Product state where the chatbot expect not commands but only a raw query from the user. The queries in

this state will be processed as it was discussed in the Product Search section before.

2) *Bot messages*: According to the intention of message that bot send to user the format of the messages can be different.

- *Pure text messages*: This format of messages are used to present a particular informations or to pose a question.
- *Buttons*: The intenstion of buttons has been explained in previous section. In addition in particular states of the product serch they can aim the user with most probable options. Like price range or existing colors.
- *Cards*: The card format is consitute with combination of multimedia objects(most likely Images), text and the link. It can be used presnt a product with album of its image. In addition few number of button can aid user to perform task relevent to this product.

B. States:

Sate is a vital part of the Telegram Bot API. Different states, handle scripts and commands differently. For instance, the same script or command will perform different methods and tasks for distinct states.

As an example, the “Profile” state is meant to gather information of users to help them to ease the task of product search. As an exemplification if a user intended to purchase a shirt; instead of entering required information such as gender or size on each search the bot can refer to data has been stored in its profile database.

However, without providing these types of information by user, it is still possible for user to make the search. Also, a user can change features in process of searching by performing the proper command. So, the profile information remain the same as before on the database. For instance, it could give the chatbot information about a product like “shirt” and add some properties like the size of “L” or “XL” she/he is looking for or else the chatbot shows all of the collected products to the user. Below, the “Shirt Size” state has been shown. In this state, the chatbot returns “Button” messages as a Bot Message, and the user is expected to press his/her desired button to send the chatbot a Command which is one of the End User Message types.

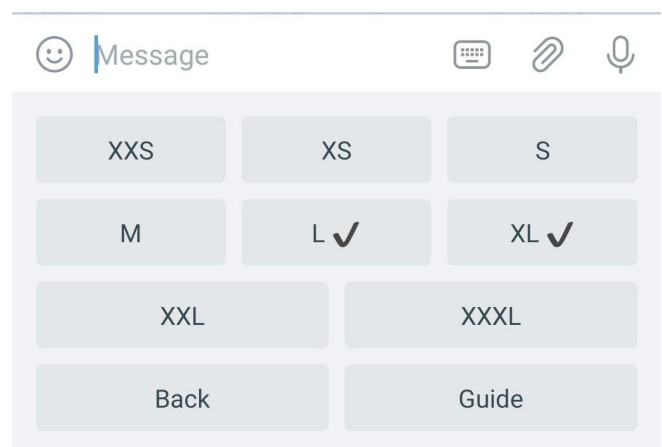


Fig. 5. Shirt Size with Multiple Choices

In the next section, we discuss some of the main methods and provide some information about messaging handlers that are used in the ShopBot. In addition, we explain the implementation challenges we encounter over the development phase.

V. CONCLUSION AND FUTURE WORK

The usability, the measure having been overlooked in many of the commercial applications. The efforts have been made to address the factors impacting on efficiency and usability of the shopping application in conversational mode via instant messaging applications; particularly, in this case Telegram has been chosen. The main achievement was the deployment of a wrapper which simply act like the other telegram users and it can send and receive messages in any form. The complete the given aim, it was mandatory to fulfill the shopping experience expectations of the users by collecting their information and taste of shopping in direct and indirect ways. Most importantly, the main objective of this project was to provide conversational interactive shopping experience which is handled with computer base agent.

For the future work of this project we could involve it with mapping the product search requests to the existing product search to improve the performance. In this case the matter of accuracy is an important factor. However even if the request is not completed and not mentioned all the properties it could be mapped to the most frequent or trending one. This module helps to reduce the task of conversation and information gathering on a product.

REFERENCES

- [1] J. Weizenbaum, "ELIZA—a computer program for the study of natural language communication between man and machine," *Commun. ACM*, vol. 9, no. 1, pp. 36–45, 1966.
- [2] L. Lin, L. F. D'Haro, and R. E. Banchs, "A web-based platform for collection of human-chatbot interactions," *HAI 2016 - Proc. 4th Int. Conf. Hum. Agent Interact.*, pp. 363–366, 2016.
- [3] N. Mhatre, K. Motani, M. Shah, and S. Mali, "Donna Interactive Chat-bot acting as a Personal Assistant," *Int. J. Comput. Appl.*, vol. 140, no. 10, pp. 6–11, 2016.
- [4] A. Pradana, "Intelligent Conversational Bot for Interactive Marketing with Consumer-centric Approach," *Int. J. Comput. Inf. Syst. Ind. Manag. Appl.*, vol. 6, pp. 265–275, 2017.
- [5] M. S. Satu, M. H. Parvez, and Shamim-Al-Mamun, "Review of integrated applications with AIML based chatbot," *1st Int. Conf. Comput. Inf. Eng. ICCIE 2015*, pp. 87–90, 2016.
- [6] S. Gupta, D. Borkar, C. De Mello, and S. Patil, "An E-Commerce Website based Chatbot," *Int. J. Comput. Sci. Inf. Technol.*, vol. 6, no. 2, pp. 1483–1485, 2015.
- [7] de Oliveira, Juan Carlos, Danilo Henrique Santos, and Mário Popolin Neto. "Chatting with Arduino platform through Telegram Bot." In *2016 IEEE International Symposium on Consumer Electronics (ISCE)*, pp. 131-132. IEEE, 2016.
- [8] Pricilla, Catherine, Dessi Puji Lestari, and Dody Dharma. "Designing Interaction for Chatbot-Based Conversational Commerce with User-Centered Design." In *2018 5th International Conference on Advanced Informatics: Concept Theory and Applications (ICAICTA)*, pp. 244-249. IEEE, 2018.