Proceedings of the International Conference on Intelligent Computing and Control Systems (ICICCS 2020)
IEEE Xplore Part Number:CFP20K74-ART; ISBN: 978-1-7281-4876-2

A Self Learning Chat-Bot From User Interactions and Preferences

Parth Thosani
Student, B.Tech Information Technology
Sardar Patel Institute of Technology (S.P.I.T)
Mumbai, India
parth.thosani@spit.ac.in

Manas Sinkar
Student, B.Tech Information Technology
Sardar Patel Institute of Technology (S.P.I.T)
Mumbai, India
manas.sinkar@spit.ac.in

Jaydeep Vaghasiya
Student, B.Tech Information Technology
Sardar Patel Institute of Technology (S.P.I.T)
Mumbai, India
jaydeep.vaghasiya@spit.ac.in

Radha Shankarmani
Professor, Information Technology Department
Sardar Patel Institute of Technology (S.P.I.T)
Mumbai, India
radha_shankarmani@spit.ac.in

***Abstract*—A chatbot is an interactive AI tool that tries to imitate human behavior that interprets information provided and responds accordingly in either textual or audio format. Nowadays chatbots are used to efficiently carry out digital communication. Our concept introduces a new field of research in chatbot communication. Nowadays most chatbots efficiently complete the required task, but there is one thing to notice. Many of the conversations between a human and a chatbot are repetitive. Technology has always been reducing human effort and hence a multi-agent system is proposed that has a chatbot as a middleware between the user and the outside world. This chatbot has something unique, i.e. it understands the user and their requirements. It is like having an assistant who understands the user & their requirements. Now since it understands the user, it has the capability of making decisions on behalf of the user, thereby reducing the efforts of a user to carry out a task. Our concept of considering one task that is done with the help of a chatbot is demonstrated. The proposed system adapts and acts accordingly on behalf of the user.**

Keywords—ChatBot, AI, Assistant.

I. INTRODUCTION

A chatbot is a system that enacts to be a human and helps the organization to complete a task for the user. Simply stating chatbots are an innovation towards Artificial Intelligence. Various organizations have adopted the use of chatbot that smartly conveys the messages and information[6]. An organization can have multiple types of users like customers, clients, vendors, logistics in fact any person who is associated with the business of the organization. All these users interact with the chatbot differently. The chatbot understands the information provided by the user and responds with the most appropriate result or carries out the required task and tries to satisfy the user requirement. There is a vast scope of all chatbots to have a human touch.

Most of the chatbots are single-ended chatbots that are trained to help users perform some tasks. The information is directly understood by the system consisting of algorithms that understand the intent of the user and the data/entities provided and will respond with a predefined response. These systems are quick and easy to use and will prevail in the world as they efficiently carry out the organizations' task, however, they sometimes lack a human touch. Most of the conversations between a user and a chatbot are repetitive and there are a lot of users' common data being shared with the chatbot, but the user still has to enter the most usual data which he/she enters everywhere, something very obvious like name, address and other primary details. With the primary details, there are many other repetitive elements like a user might be ordering the same pizza many times. These parameters are user-specific.

The proposed system understands what the user wants and accordingly saves the users' effort in completing a daily task by communicating with the second chatbot on behalf of the user. Our system tries to overcome the ignorance of user preferences and requirements in taking decisions on behalf of the user. Both the server & conversational bot interact among themselves and function as if it is a normal conversation and the resulting output is sent back to the user. This leads to either reduced or sometimes no input from the user while adapting the preferences and likings of the user. User preferences play a vital role in the conversation. The more the user interacts with the system, each time it trains the system to understand the user preferences. The recency effect is showcased while responding to the second chatbot by considering the past few conversations made by the user.

Automation of a chatbot communication such that it communicates on behalf of human considering preferences is what has demonstrated as a proposed system. By

self-learning, the system is adaptive and takes into consideration every preference of the user and learns more and more about the user over time.

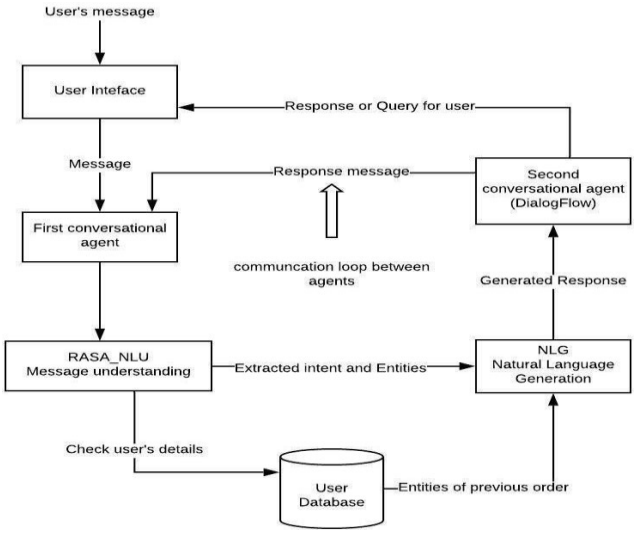

Fig. 1. Flow diagram of the system

II. LITERATURE SURVEY

Our system takes into account the magnificent work done by RASA which is thoroughly explained in [1]. The sentence which is given by the user is usually broken down into intents and entities by using RASA's Natural Language Understanding (NLU), which converts it into a JSON format and hence the values can be extracted easily. Various intents are given a parameter which is in percent named as 'confidence' for intents, which specifies the probability of matching to that defined intent. Also for entities, there are various parameters namely 'confidence', 'value', 'extractor' to name a few important ones. But, without containing any front-end for specifying the intents and training makes it a bit tedious, so the system along with the Dialog Flow is used.

Reference [2] explains the collaborative reinforcement technique which is used by the two conversational agents, which is always purely done by NLG and NLU, so apart from just accurately generating and understanding the text, there's a need to reduce user's load by providing an alternate mechanism to remember the preferences and history of the past conversations.

Reference [3] is about single conversational agent where user can ask a question (query) to the chatbot and chatbot will use Natural language Processing (NLP) to process the message and identify the intent of the query and after that, it will check in the data which stored in the database and try to give a response to the user and resolve the query of the user to improve this, used multiple conversational agents[10] so that the query asked by chatbot can be minimized and we made the query resolution of the user more efficient.

Reference [4] talks about how two conversational agents can communicate with each other and learn concurrently without a simulated user which will minimize the user interaction and agents will learn while communicating with each other. Reference [5] talks about the use of a web socket connection and its benefits over a normal https connection. The paper shows results that in a web socket connection the payload is much lesser than for an https request and the time required for the communication is significantly lesser.

Reference [6] talks about how chatbots play a vital role in our lives and how they replace humans from their jobs. They have examined how the chatbots have evolved and analyzed what could be the future of chatbots per se. A chatbot's job is to help a user communicate with a computing software in an unstructured manner, the way humans generally communicate. Day by day the chatbots are getting more efficient in providing seamless service in different domains. This evolution of chatbots leads us to make a conversational agent, currently restricted to one domain, acting as a middleware to further reduce human efforts in doing tasks.

Reference [7] talks about Goal-oriented chatbot that is trained on a particular domain for example food ordering chatbot. In this paper, a chatbot is trained on domain-related data using Deep reinforcement learning so that the intent of the user's message can be identified and to generate a response to the user's message Recurrent neural network is used in this paper. Reference [8] talks about Natural Language Processing of human-written text which includes Language understanding and language generation and components involved in NLP that is Phonology which refers to sound, Morphology which is word formation and structure of the sentence, Semantics syntax and Pragmatics is understanding of the text for intent and entity identification. Reference [9] talks about how natural language generation works and how Recurrent neural networks are trained on a particular domain to generate appropriate response according to the context which is achieved by knowledge of the domain, knowledge of the language, strategic rhetorical knowledge.

Reference[13] talks about the capabilities of chatbots and how they are providing efficient and seamlessly fast communication, and nowadays how simple it has become to build a chatbot using machine learning and how easily it can be deployed/integrated with any messenger/assistant service.

Reference [14] talks about how chatbots are nowadays used in big corporates like Amazon and Microsoft as an efficient technology to communicate with their stakeholders and also with their customers for resolving their queries with effective communication.

III. PROPOSED SYSTEM

The specific domain for this model is a pizza ordering system. Initially, the user is new and has not made any previous orders on the system and hence cannot learn the preferences of the user. The chatbot understands the requirements of the user once there are at least 5 orders made, the server which tries to enact the user, takes in the requirements and necessary basic details. The user has a choice to either make a completely new order or the user wants the system to order on his/her behalf using the past orders. The server provides both REST APIs and a socket connection to the clients for the basic functionalities & the chat application respectively. REST APIs in NodeJS are asynchronous i.e. they do not make the client wait for the response. While a Socket Connection is one that provides a real-time full-duplex connection between the clients and the server[5]. Once a REST API responds to the client back, the server has no track of the client to access it. A socket connection is necessary for the implementation of a chatbot.

The entire system is divided into 3 different parts:

A. Ionic (Front-End)
B. Server & Conversational Agent (Middleware)
C. MongoDB Database (Back-End)

A. Ionic (Front-End)

This step is the beginning of working of the system, the user logs into the system by providing basic details and information. As the platform for the development of the front-end is ionic, which is a cross-platform mobile application development framework that is built on top of frameworks like Angular, React, VueJS and pure Javascript which gives a developer the freedom to use the tool efficiently. The user is shown the basic interface of a chat system, where the user inputs all the queries regarding the pizza and can order by building pizza of his own choice. All the queries or questions are put forward by the user by simply entering the keywords or a series of words that are interpreted by the system on what the user wants. On the implementation part, required a socket connection with the server that listens and emits required events for the chat to take place. Other functionalities that do not require a real-time connection are taken care of using RESTful APIs. For making & using the socket connection, used‘ngx-socket-io’ which is an npm module. On the Security front, used JSON Web Token(JWT) which makes sure secure communication between the server and the client while providing authorization of functionalities at the server.

B. Server & Conversational Agent (Middleware)

The main logic of the functioning of a chatbot lies in this part of the system and take minimum requirements from the users and provide them with the most appropriate and matching pizza item. In the case of a new order, the user has to provide the specifications of the Pizza required, in this case, the server directly makes the user communicate with the 2nd Chatbot.

Once the conversation is over the entities are extracted from the session and then stored in the database for preference analysis on further orders. In case the user opts for the system to talk on his/her behalf, the system uses the prior orders made by the system and by analyzing them makes a preferred order. Now the system/server/first conversational bot uses this order and communicates with the second agent. The conversation is carried out between the server & the second agent until all the parameters are filled, the resulting value is generated by the server and then given back to the user.

1. *Server*: The server plays a vital role in the working of the system. The main aim of the server is to understand the user’s message and react accordingly for that using RASA because when it comes to message understanding there are so many NLP stacks from cloud AI incumbents but in this kind of cloud AI model is trained on a conversational flow chart (Predefined conversation flow between user and agent) which is not efficient for building a good conversational agent but when it comes to RASA it uses probabilistic models and reinforcement learning for model training and for controlling the flow of conversation. Also, it is an open-source NLP & NLU model thus providing the flexibility for modifications. Because of that, it doesn't face problems like lack of training data, out of vocabulary words, similar intents due to this it is more reliable to use for Natural language understanding. RASA NLU provides a model that has to be trained by providing various types of entities and intents. The model is thoroughly trained and then is ready to understand the language. The goal of the server is to understand what the second agent is asking for. For this, the model is trained with various defined intents that the second agent would respond with. After using NLU, this agent then generates the text that has to be understood by the second agent.

The NLG or Natural Language Generation is used by the server to pass on the parameters to the second agent and also can question the second agent about the pizza types, toppings, etc. For generating a response based on the intent of the user’s message and entities are using ‘nlglib’ of python which can generate clauses with the help of NP and VP. NP will be used for generating the noun part of the response and VP will be used to generate the verb part of the sentence based entities extracted from the preference order. This NLG is done and only important parameters and values are passed, so better results and accuracy is achieved.

Another job of a server is to check the database frequently and update the values of the order items and the variables and entities. The user can then order the same item, or can just tell the system to order based upon the past orders made i.e.

preference of the user. This prevents the users to give in the details of the pizza whenever they wish to order, the server works on behalf of the user and classifies his/her requirements without even actually wasting any time of the user. The preference analysis for the user considers all the paste orders of the user to give the best results. This reduces the efforts of the user and serves the purpose of ordering a pizza more efficiently.

On the implementation front of it, used NodeJS to provide an HTTP server created using express.js and also using socket.io to get a socket connection provider on the same server. The basic functionalities on the server like authentication and other database queries are carried out using asynchronous RESTful APIs. For the chatbot, events are managed using socket.io on the same server. The functionalities that require python i.e. NLG, NLU, etc. The 'python-shell' is used which is an npm module that is used for spawning python child processes for executing the required python scripts that use the models and provide us with the output required. The socket events are such that their callback functions manage the flow of the whole conversation between the first and second agents in case of preference orders.

2. *Second Conversational Agent*: The second agent contains all the items and pizza details which the users request. The sentence generated from users via the server which is filtered in a way just to understand important parameters and entities is given to the second agent. The basic entities created for the pizza order system are [Pizzas], [Base], [Toppings], [DietPreference], [Size], [AddOns], [Extras]. The second bot understands the intent and entities and checks in the system the matching information and order details. The best and most appropriate option is selected dynamically based on the information gathered and understood. For the pizza ordering system, and have considered DialogFlow as the second bot[11]. Various intents and entities similar to the server are declared for the second agent. The process of question-answer begins between both the agents once the conversation is initiated by the user by providing the first sentence of the query. There is a series of information flow between the server & the bot and finally, the output is given back to the user.

Consider if the second bot asks the server about the information not provided in previous conversations, then the model or server fetches the information in the database or directly this question is sent back to the user and whenever the user replies, the same process of filtering the sentence is done and given back to the second agent. In this way, a three-way conversation is established to optimize the requirements. The creation of the system is in such a way and it is transparent such that the user feels he/she is talking to an agent from the pizza store/restaurant whereas they don't have any information about the server & the agent. Consider if second agents ask for the location of the user to the server, the server is trained in a way that it need not ask the user about the location and it fetches on its own making least interaction with the user and serving the needs of the user without any hassle.

C. MongoDB (Database)

MongoDB is a NoSQL Object-oriented Database Platform which shows much better & faster query results than a SQL Database while providing the developers the freedom from Schema and Complex Relations[12]. The users are stored in one collection of the database, the orders in the other. The data is then queried and used by the server. In case of Preference Calculation of the user, consider all past orders and sort each entity by usage, i.e. if a user has ordered Pizza 'Peppy Paneer' for the most number of times, then that becomes the preferred Pizza for the user and similarly, all entities are calculated.

IV. RESULT

Result of RASA NLU:

When the message is passed to the RASA NLU model it will identify the intent of the message and extract entities from it with the value along with confidence percentage in intents and also in entities along with the type of extractor.


```
"text": "i want veg pizza with onion as toppings"
```


Fig. 2. The sentence provided by user


```
"intent": {
    "name": "itoppings+idietpref",
    "confidence": 0.9601914882659912
},
"entities": [
    {
        "start": 7,
        "end": 10,
        "value": "veg",
        "entity": "DietPrefrence",
        "confidence": 0.6091555162576161,
        "extractor": "CRFEntityExtractor",
        "processors": [
            "EntitySynonymMapper"
        ]
    },
    {
        "start": 22,
        "end": 27,
        "value": "onion",
        "entity": "Toppings",
        "confidence": 0.9400513496072639,
        "extractor": "CRFEntityExtractor"
    }
],
```


Fig. 3. Intent classification done by RASA NLU

Result of NLG:

Based on the intent 'QDietpreference' identified in the JSON object NLG will generate an appropriate response to the user's message. Fig 4 shows that RASA NLU classified the intent and hence passing the value to NLG, a sentence was formed. The NLG was done on the basis of the Noun Phrase and Verb Phrase part wherein the dynamic values are given as an input i.e. "Veg".

```
PS C:\Users\Hp\Desktop> python nlg.py
QDietpreference
Diet preference is Veg.
PS C:\Users\Hp\Desktop>
```

Fig. 4. NLG which generates sentence

Traditional Order made by user:

Normally the user would communicate directly with the agent in the following flow. Fig. 5 shows the flow of the normal conversation between user & the agent.

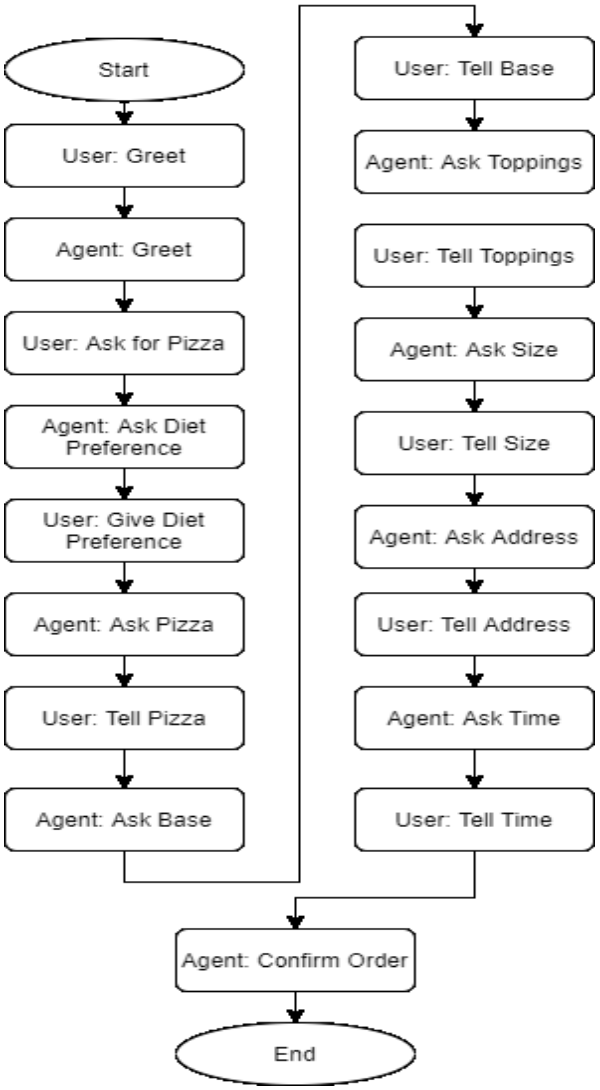

Fig. 5. Conversation between user & agent

Order made through our system:

When the agent is involved, the data that the bot can provide is easily provided by the bot, otherwise it asks the user for the information required. Fig. 6 shows the flow of the conversation handled by the bot. Agent 1 is the one that knows about the user and can provide information, whereas agent 2 is the one organization's agent who asks for required information. The bot comprises both these agents.

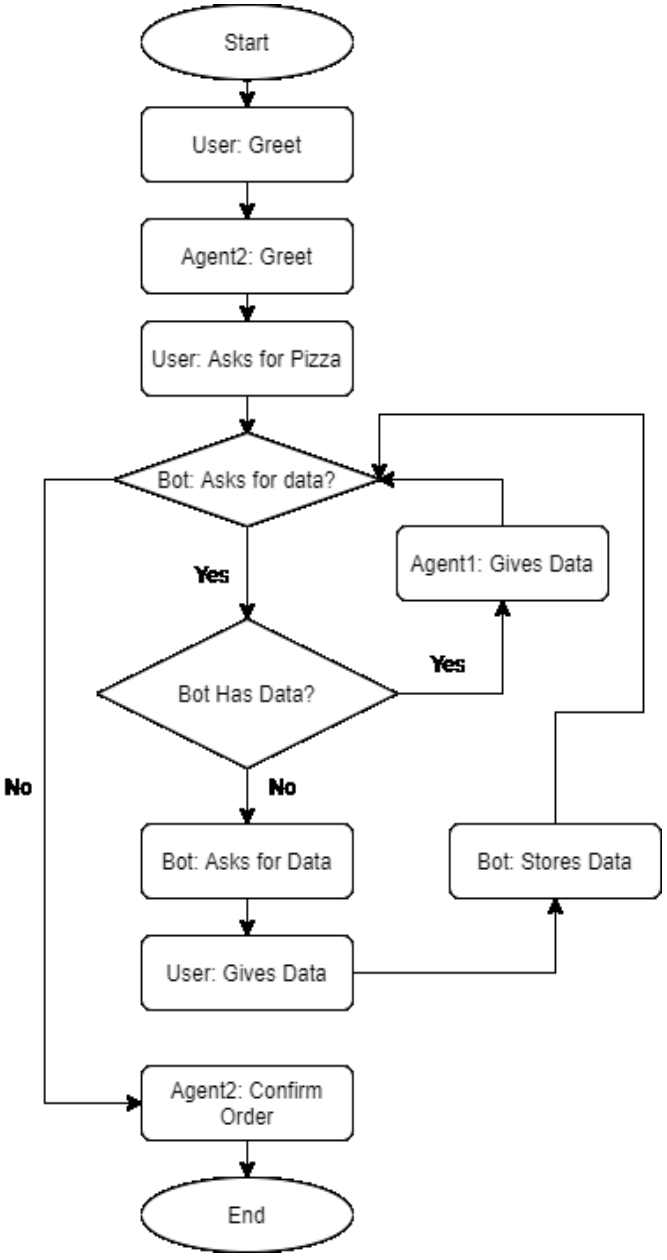

Fig. 6. Conversation flow when the middle agent is involved

Order stored in database:

The orders are stored in MongoDB in NoSQL format.

Fig. 7. Storing of order into the database

V. CONCLUSION

In this paper, a novel approach is proposed for the application of chatbots. Our system is a proof of concept. The system model adapts to the user preferences and accordingly communicates with the second chatbot thus reducing the users' messages. Normally there are at least 9 messages a user will have to send to order a pizza including the address. Our system demonstrates that based on the past orders the user has made, the user can allow the system to order on their behalf that already knows the users' address. So the user need not waste time thinking of what pizza to order and where to deliver it as the system already knows what the user likes best. Hence the purpose of the demonstrated system is satisfied.

VI. FUTURE SCOPE

The system can be adapted to various types of inputs from users to perform tasks. The chatbot can learn more about the user and that will help it to make the decisions the user would possibly make in a given scenario. The system can be enhanced by implementing reinforcement learning mechanisms and then can be used to serve multiple purposes and have tried to provide a proof of concept but it can be improved more by adding an AI Layer to the system. The AI layer is not simple but for our consideration, kept it simple.

VII. REFERENCES

[1] Tom Bocklisch, Joey Faulkner, Nick Pawlowski, Alan Nichol, "RASA: Open Source Language Understanding and Dialogue Management"

[2] Alexandros Papangelis, Yi-Chia Wang, Piero Molino, Gokhan Tur, "Collaborative Multi-Agent Dialogue Model Training Via Reinforcement Learning "Uber AI San Francisco, California.

[3] Aafiya Shaikh1, Dipti More2, Ruchika Puttoo3, Sayli Shrivastav4, Swati Shinde4, "A Survey Paper on Chatbots "International Research Journal of Engineering and Technology (IRJET) Apr 2019

[4] Kallirroi Georgila, Claire Nelson, David Traum, "Single-Agent vs. Multi-Agent Techniques for Concurrent Reinforcement Learning of Negotiation Dialogue Policies", University of Southern California Institute for Creative Technologies 12015 Waterfront Drive, Playa Vista, CA 90094, USA

[5] Liu Qigang, Xiangyang Sun, "Research of Web Real-Time Communication Based on Web Socket" International Journal of Communications, Network and System Sciences 05(12):797-801 January 2012

[6] H. N. Io, C. B. Lee, "Chatbots and conversational agents: A bibliometric analysis" 2017 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)

[7] Vladimir Ilievski1, Claudiu Musat2, Andreea Hossmann2, Michael Baeriswyl2 "Goal-Oriented Chatbot Dialog Management Bootstrapping with Transfer Learning "1 School of Computer and Communication Sciences, EPFL, Switzerland 2 Artificial Intelligence Group - Swisscom AG

[8] Diksha Khurana1, Aditya Koli1, Kiran Khattar 1,2 and Sukhdev Singh1 "Natural Language Processing: State of The Art, Current Trends and Challenges" Manav Rachna International University

[9] John Bateman, Michael Zock, "Natural Language Generation(Cognitive, linguistic, social dimensions)", French National Centre for Scientific Research

[10] Zojan Memon, Akhtar Hussain Jalbani, Mohsin Shaikh, Rafia Naz Memon and Ahmed Ali "Multi-Agent Communication System with Chatbots" Mehran University Research Journal of Engineering & Technology

[11] Antoine Raux, Yi Ma, Paul Yang, and Felicia Wong "PizzaPal: Conversational Pizza Ordering using a High-Density Conversational AI Platform" Empirical Methods in Natural Language Processing (System Demonstrations), Brussels, Belgium

[12] Győrödi, Cornelia & Gyorodi, Robert & Pecherle, George & Olah, Andrada. (2015). A Comparative Study: MongoDB vs. MySQL. 10.13140/RG.2.1.1226.7685.

[13] Nath M.P., Sagnika S. (2020) Capabilities of Chatbots and Its Performance Enhancements in Machine Learning. In: Swain D., Pattnaik P., Gupta P. (eds) Machine Learning and Information Processing. Advances in Intelligent Systems and Computing, vol 1101. Springer, Singapore.

[14] Lalić D., Stanković J., Gračanin D., Milić B. (2020) New Technologies in Corporate Communications. In: Anisic Z., Lalic B., Gracanin D. (eds) Proceedings on 25th International Joint Conference on Industrial Engineering and Operations Management – IJCIEOM. IJCIEOM 2019. Lecture Notes on Multidisciplinary Industrial Engineering. Springer, Cham.