IRSTI 28.23.35 https://doi.org/10.26577/JPCSIT.2023.v1.i2.03

D.E. Kyzyrov * , **S.Z. Sapakova**

International Information Technology University, Kazakhstan, Almaty
*email: daniarkk00@gmail.com

CHATBOT ASSISTANTS: IMPLEMENTATION AND ANALYSIS OF THE EFFICIENCY

Abstract. Nowadays, chatbots are used only on a few websites. People can't always adapt to the design of every website, so they need an assistant for that. On every website, a chatbot can find its calling, and it can be used for different needs, and for solving different problems. One of the main problems on the Internet is copyright infringement, copying other people's sites. Various fake websites copy official sources, and these types of fraud disguise all sorts of links as normal official links, and people often fall for this deception. Such websites may also use chatbots for their evil designs. By spreading scam links using chatbots, scammers can gain access to users' data, social media accounts, bank card details, and other important data. Therefore, there is a need for a thorough study of chatbots, as well as their correct and effective use. A chatbot is a multifaceted product that can be used in any area of life, from simple information searches to education and medicine. This article will present some types of chatbots, as well as information on where they can be used and for what needs. The purpose of the article is to analyze the effectiveness of chatbots and also offer some methods for determining the effectiveness of using a chatbot. There will be information about the first chatbots, about the trends of our days. Also in this article, a functioning chatbot model will be presented, which will be used in one of the most popular communication platforms, Discord.

Key words: Conversational Agent; Machine Learning; Natural Language Processing; Neural Network; Internet of Things.

Introduction

Due to the coronavirus pandemic and the development of Internet services, the number of people who use web surfing has grown tremendously. Therefore, fake chatbot assistants are more effective in fraud than ever. Fraudsters began to actively create fake web pages and pass them off as official. This method of scamming by using a chatbot is very effective from the scammer's point of view, as the victim may simply not notice the letters or symbols that have been changed to disguise the suspicious site. Take the film industry for example. Fake chatbot assistants can be used to give an incorrect link to a particular movie. Their links may differ from the official links to movies or TV shows, but the average user may not notice this.

Usually, chatbot assistants are most used on social networks. As an example, we can take the social network Vkontakte. The number of people who use one chatbot on the Vkontakte social network today ranges from ten thousand to 1.5 million users. Such many people can become a source of income not only for the field of entertainment [1] but also for scammers. If the problem of fraud with such fake chatbots is not solved, then people will simply stop trusting even the official entertainment distributors.

Unfortunately, chatbot assistants are not used in all official sources. In the film industry, in such official distributors as Netflix, Disney+, and others, there are no chatbot assistants that could recommend films at the user's request or select films for him. Therefore, the question remains open whether it is worth creating such chatbots for people.

To avoid problems with scammers and fake chatbot assistants, developers should contact official sources directly to create chatbot assistants. After all, if the chatbot assistant is associated with an official source, then there can be no fraud from third parties. Because chatbots will only provide official links to movies and TV shows.

Also, the recommendation system can be linked to the most authoritative sites with reviews or movie ratings. For example, sites IMDB, Rotten Tomatoes, and so on. Here people write real reviews and movies with TV shows are rated not only by ordinary users but also by professional critics. Therefore, by linking the chatbot to an official source, the film industry will not lose any income, and ordinary users will be far from scammers.

As for the development of a chatbot itself, it can be created from scratch, by the developer himself, or with the help of special tools for creating bots. Such tools can help not only developers but also those people who had no experience in development.

Chatbot assistants can also be used in all messengers and social networks. If you do everything officially and get rid of suspicious chatbots, there will be less fraud, and everyone will only benefit.

Related work

Chatbots are used all over the world by different people for different reasons. Looking at the graph, you can understand that the term chatbot appeared already in 1966 [2] and became very popular after the 2000s as it is shown in Fig. 1. The idea of communicating a computer with a person came to Alan Turing's head already in the 1950s, after which, the first chatbot called ELIZA [3] was created.

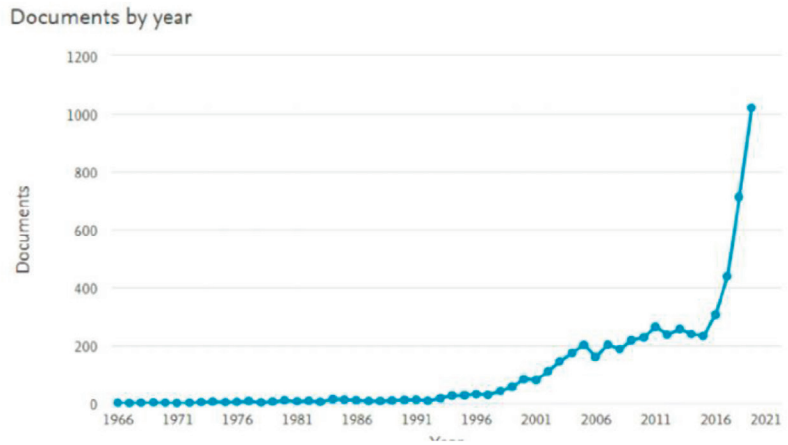

Figure 1 – Search Results from Scopus (Chatbot keyword by Year)

The chatbot keyword [4] has the highest usage in the United States by country and the lowest rating in the Netherlands as it is shown in Fig. 2.

Chatbots are a versatile product that has many types and uses. As it is shown in Table 1, they can communicate with you through text, photos, or voice, they can also have different goals, permissions, and services [5].

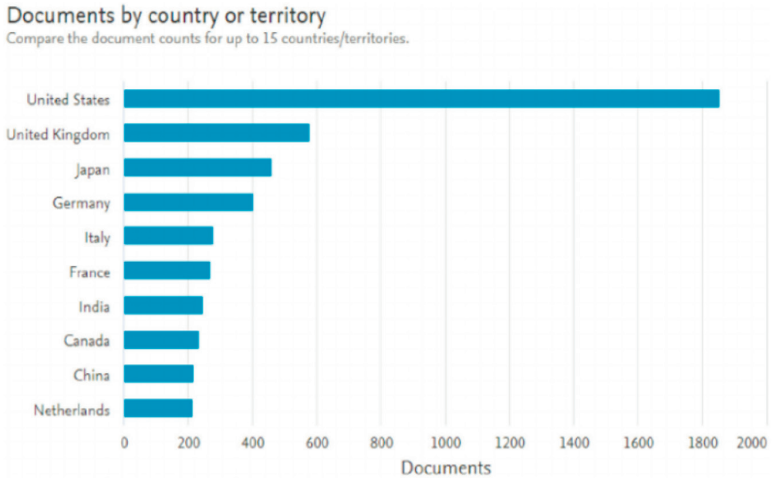

Figure 2 – Search Results from Scopus (Chatbot keyword by Country)

Table 1 – Chatbot Classification

Knowledge Domain	Generic, Open Domain and Closed Domain.
Service Provided	Interpersonal, Intrapersonal, Inter-Agent
Goals	Informative, Chat-based, Task-based.
Response Generation Method	Rule-based, Retrieval-based, Generative.
Human-Aid	Human-mediated, Autonomous.
Permissions	Open-Source, Commercial
Communication channel	Text, Voice, Image

Several reasons for using chatbots have been identified [6] based on a study of articles related to chatbots:

Chatbots for productivity: The main reason for using them is their simplicity, convenience, and speed of use. Chatbots increase user productivity. With the help of chatbots, you can access a lot of information.

People appreciate chatbots when they save them a lot of time, or when they make it easier or faster to get information [7], for example, by providing efficient assistance in a customer support situation or by pointing to an easy-to-use manual or FAQ.

Sometimes people use chatbots as an answering machines, when the other person writes to them, people set up chatbots so that they give answers to basic questions. In this case, they do not have to waste time answering simple questions and for this they use chatbots.

To obtain help and information: As we said, the availability of information and the ease of obtaining them is one of the reasons for using chatbots.

Some people use chatbots as useful tools in their daily online activities. They use them as a recommendation system, for example, when a person likes to travel and needs recommendations or advice, a chatbot can easily find an interesting place and give useful advice.

Also, some people prefer chatbots in the role of helpers to real people, because they are shy about people. They will not need to feel uncomfortable and afraid when asking silly questions to the chatbot.

Chatbots for entertainment: Some people use chatbots for fun, like just chatting with them and enjoying their responses. People ask them questions and amuse themselves with their funny answers to all sorts of questions. Some people negatively use these chatbots to just kill time and waste it.

After all, chatbots are very easy to use, with their help of them you can easily find the movie, game, or music you need. It will be enough to write the name, and then the search system will give you a message in response with a link to your request.

Chatbots for social and relational purposes: Chatbots are also good conversationalists, some chatbots are specially designed to communicate with them. They collect information from all messages that they write, and with the help of machine learning, they improve and become good interlocutors who can answer any of your questions.

According to data from the site botostore.com as it's shown in Fig. 3, there are more than 120,000 different chatbots on the social networks messenger.com and Telegram today. Each of the bots has a certain number of parameters, language of work, and response time. Such bots are used in different whole and for different areas of life. They can be used to find information, photos, recommendations, a personal assistance, and a bunch of other purposes.

A chatbot can have many roles and functions as a salesperson on your website, provide sales services, generate invoices, pay for an order, or track its status, it can be seen in Fig. 4. The bot can also integrate with the corresponding APIs to fulfill client requests, provide them with authorization, integration with mobile or web services, and after integration, issue the result in the desired format. The bot can also act as an information provider or helper, provide automation of processes on the website, for example, demonstrate the rules, introduce new participants, or advertise something.

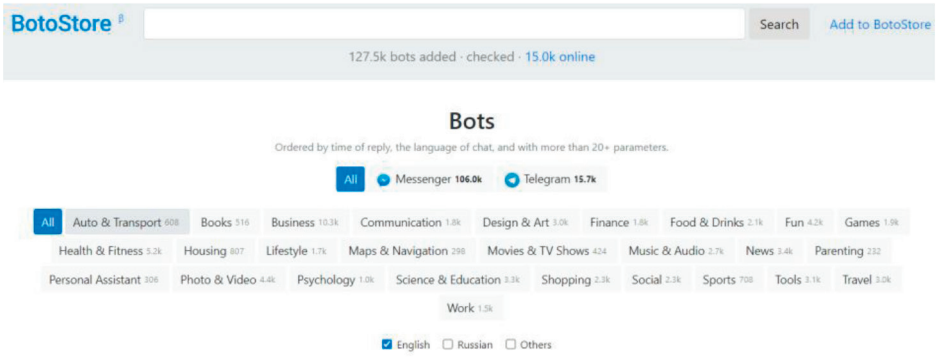

Figure 3 – Chatbots from Botostore

According to the statistics of the site for creating chatbots aimylogic.com, chatbots can answer 80 percent of frequently asked questions if they are properly trained.

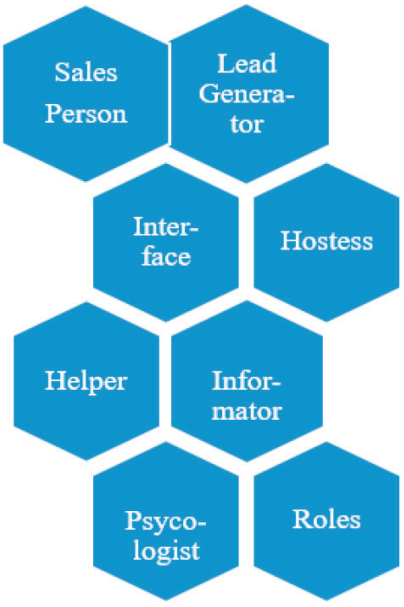

Figure 4 – Chatbot Roles

There are several types of chatbot technologies. Depending on the type of technologies used chatbots can be Rules-Based and AI-Based [8].

Rules-Based chatbots adhere to the original creation rules as it's shown in Fig. 5. At the very beginning of their creation, development scenarios are set, and chatbots will work only in the direction where the developer programmed them and the chatbot will not be able to deviate from them. Accordingly, chatbots will not answer questions from other scenarios.

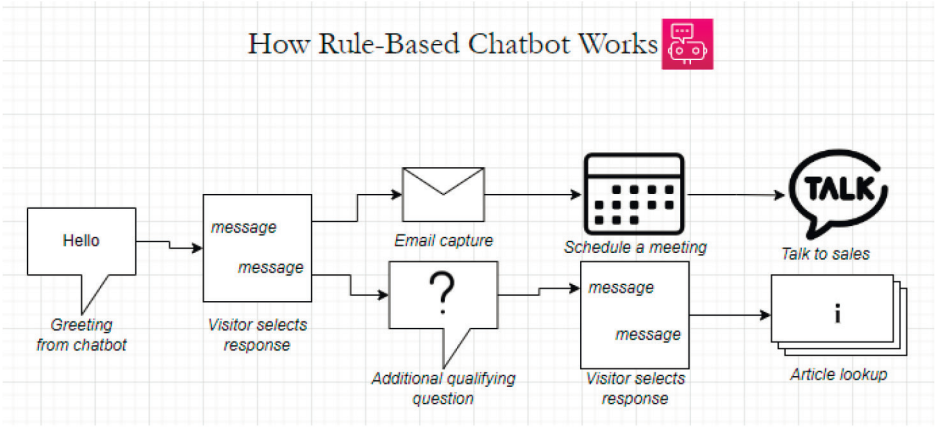

Figure 5 – Rules-Based Chatbots

AI chatbots are initially trained by the developer, but then they can automatically learn after training. The functionality of such chatbots is more advanced than that of rules-based chatbots. As it's shown in Fig. 6 chatbots can understand user responses and requests, but if the user is not satisfied with the response, and the request was not found among the intents, then the answer was given incorrectly. The self-feeding system [9], that is, the Artificial Intelligence [10] of the chatbot can help here. It will allow the chatbot to generate a response from the user's feedback [11] so that the next time the chatbot can give the correct answer.

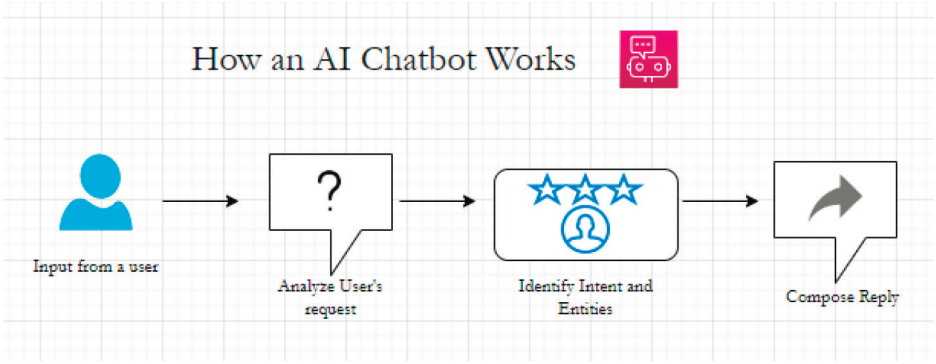

Figure 6 – AI Chatbots

As we said before, chatbots can be created from scratch or by using chatbot builders. Chatbot builders are development platforms where you can create and modify and customize chatbots depending on your needs. Some chatbot builders do not require programming knowledge, such builders are the best in their field since everyone can create their chatbot.

Chatbot builders can usually integrate with email systems, CRM, and other automation tools that will make your sales, marketing, and other processes more efficient.

Also, chatbot builders usually have built-in templates, by customizing which, you can create a chatbot for yourself in the shortest possible time.

In terms of their effectiveness, chatbots with built-in machine learning and artificial intelligence stand out the most. Because they will only improve over time.

Here are some chatbot builders you can use to create your first chatbot:

Giosg: Here you can create a chatbot without programming knowledge, many templates are ready

to use. The site also has many options for customization, as well as integration with a live chat, email integration, a free trial period, the ability to request a demo, etc. They also have a guide on how to build chatbots.

Nine Easy Steps to Build Chatbot: Decide the type of chatbot; Determine KPIs; Understand User Needs; Give the bot a persona; Plan the bot flow; Design the bot; Preview and test; Target the bot; Measure and Optimize.

MobileMonkey: One of the most popular chatbot options. This builder is multi-platform because through it you can create a bot for Facebook, Instagram, SMS, and websites. It also provides an opportunity to analyze your chatbots, but compared to the previous one, here you already need to have some knowledge in the field of programming, that is, not every person can use it.

Chatfuel: This is a chatbot builder in which you can create a chatbot for Instagram and Facebook through the drag-and-drop system. You can create a simple FAQ bot in a matter of minutes, since the platform does not require coding skills, anyone can use it.

The Internet of Things is a technology that shapes the digital age, it connects intelligent objects or things that are developed based on Internet architectures and can interact with each other to achieve a common goal. The main advantage of the interconnection of such sensors, mechanisms, and services, as well as the collection and processing of data from them, is awareness of various situations and the ability to provide users with an understanding of themselves and the world around them. We can see the Internet of Things in our daily lives, in the field of medicine and healthcare, manufacturing and agriculture, home automation, building automation, energy management, and other "smart devices". Typically, IoT developers connect to the device through cloud services such as IBM IoT, Microsoft Azure, Cisco IoT, AWS IoT, and others.

The main reason for the development of the Internet of Things is the spread and use of the Internet in our daily lives. We use the Internet wherever possible to access information, for entertainment purposes, to watch media, to communicate, to communicate with friends through social networks, and much more. Currently, according to Google, about sixty percent of the world's population uses the Internet through their devices, mobile phones, computers, and so on, which is approximately 4.7 billion people, and this number is increasing every day. Therefore, we can say that the Internet has an important role as a backbone for the exchange of information and the relationship of people with computer technology and its networking capabilities. However, the Internet alone cannot solve all the problems associated with the Internet of Things. Let's find out what problems can arise in the field of the Internet of things.

Challenges in IoT: One of the problems with the Internet of Things is that they operate in isolated technologies and vendor repositories, which in turn limits its capabilities, functionality, and interoperability and creates a kind of incompatible area. In particular, it limits the communication of dissimilar devices, such as household appliances, mobile phones, sensors, and other devices.

Secondly, a large number of connected devices are already starting to create management problems, in particular with the management of devices, applications, and data from them. Market giants such as Microsoft Azure, IBM IoT, Cisco IoT, and AWS IoT offer cloud services to solve these problems so that enterprises can quickly connect to services or software. But even if such correct solutions were provided, some problems remain, especially with interoperability, which in turn can lead to fragmentation. Also, IoT systems are having difficulty unifying user interfaces. Many users find it difficult to access applications and information for every new object in the IoT ecosystem.

Problems in the field of the Internet of Things [12] can be divided into two main types. These are problems based on technology, and problems at the center of which is a person.

One of the most important problems in the field of the Internet of Things, which is a technology-based problem, is the problem related to big data. The data that various sensors generate is not only large, but it is also diverse since data is taken from sensors for temperature, sound, light, video, and so on. Obtaining data from sensors, and the ability to present them in an understandable form and on time is a very important and difficult task. This problem is well illustrated by the knowledge hierarchy – DIKW (Data, Information, Knowledge, and Wisdom). As the hierarchy moves up, the data gets smaller, but getting the knowledge needed to gain intelligence (Wisdom) from it becomes more difficult.

There is a problem that is directly related to the transfer of data to the user clearly and simply. This is where a chatbot can be used to understand the true intent of a user request. They can be programmed to be accessed from the Internet and process requests in real time. Information in IoT panels is often

overflowing with various data, graphs, charts, and tables, which makes it difficult for users to access the information they need. Chatbots can be used to get answers with accurate information. They also can simplify information from various big data and graphs into a simple conversational interface to easily maintain a dialogue with the user.

One of the problems in the Internet of Things is fragmentation. When one device is difficult to access between dissimilar devices. For example, when there are heating, ventilation, or air conditioning systems in the building, they have different control terminals. Chatbots in such cases can become a kind of connecting device that will serve as a single interface. It will be enough to give permission from all devices to manage for the chatbot.

Devices that are subject to the Internet of Things can track health or can transmit information from industrial sensors, and here monitoring and reporting are key aspects. The interconnected use of a chatbot and Application Performance Management can improve monitoring efficiency as well as increase reporting by providing data to users in a timely and efficient manner.

Chatbots are assistants that were created to facilitate the relationship between a computer and a person, so it can be said that the effective interaction of a person with the Internet of Things system is also the main task for chatbots.

In today's world, the Internet of Things is developing very quickly, in the course of this, more and more functions are appearing on devices, and options for using these same devices are increasing. Accordingly, there is an additional demand for end-user training. And this, in turn, will be a difficult process for both developers and ordinary users. This process can also be simplified using chatbot assistants. For example, add a feature to provide usage instructions for a chatbot that would help users understand how to use a particular device. So, chatbots could recommend what decisions are best to take, and which action will be more intelligent.

Chatbots are also good for automation. Sometimes there are problems with solving tasks of the same type, where cyclic actions are required from the user. With the help of regular API calls, the chatbot would be able to solve such issues on its own.

Methods and Materials

Effective chatbots can be called chatbots that resemble people as much as possible. After all, such chatbots give answers based on real messages, and give answers clearly and in detail. These chatbots will be able to fulfill any role on your website. But it is also worth considering that making any mistakes from such chatbots will lead to a large decrease in efficiency. Because the expectations of the users will be too high and, accordingly, the disappointment will be quite noticeable.

You can also check for effectiveness by the number of users involved. These are the users who came into contact with the chatbot. This data is measured as a percentage of the total number of people who visit the site or application. But for call centers or other services to users, such a metric will not be relevant.

The number of active users. These are users who wrote to the bot on their own, and not in response to its messages. This metric reflects the most important efficiency of a bot – it's utility. If people write to the bot themselves, it means that the chatbot service is important for people.

The number of repeat users. The previous two metrics show the effectiveness of a chatbot very well. But often such users can come into contact with the bot one single time. Sometimes this indicator is good, as it can mean that the question that was asked to the chatbot has been resolved. But if you wanted the chatbot to be contacted several times and regularly, then a decrease in such a metric is a reason to reconsider the structure of your chatbot.

If a chatbot is used to send messages, then the Number of Messages Read can be another good metric for you. Since you will be able to send notifications about any promotions, discounts, etc. A good chatbot has average readability between 40 and 70 percent.

You can also take the Average Session Duration. This indicator can have different results for different chatbots. If you use a chatbot to provide services to users, or use the chatbot as an assistant, then this metric should be short. Because your chatbot needs to respond to messages quickly and clearly. In case your chatbot is used for entertainment purposes, then the duration of use should be long. The longer a chatbot is used, the more efficiently it works to entertain users.

As we said, many chatbots use NLP [13] technology. When a chatbot understands ordinary human language and can parse it, the chatbot becomes very efficient. However, one hundred percent result in the world has not yet been achieved.

Chatbots may not understand people or be mistaken in their answers. There is nothing wrong with that. It is important to use the data from these errors

correctly. Use the data from these errors for the further development of the chatbot. So, the effectiveness of the chatbot will be one level higher.

There is also a Goal Compliance Rate indicator. A unique metric that is used to measure the achievement of goals set by the user himself or the creator of the chatbot. This means that you should measure the number of completed requests and requests that were stopped halfway.

All of these metrics are quantitative. In addition to them, semantic analysis of user replicas is also used. This helps to understand the emotional background of the conversation, find out which queries are the most popular, which queries are of value, and get a huge amount of other information.

Chatbot Implementation

The chatbot was created using the Python programming language, and designed for use on the Discord platform. The Python programming language was chosen because it is the most popular today and no less effective, affordable, convenient, and easy to create a chatbot. It provides a lot of libraries for working with NLP and machine learning, which is why it will be advisable to use Python to create a chatbot with an intent.

Discord is one of the most popular communication platforms, especially in the gaming community, but it is also popular among other communities. The platform allows you to join many servers, create your servers, and communicate via video, audio, or text. Here you can create unique roles, and channels, access certain functions, and integrate the server with many other applications.

The chatbot is based on the Discord API and many Python libraries, such as TensorFlow, NLTK, NumPy, etc.

First of all, we go to the official Discord website, then the Discord developer portal. Next, go to the Applications tab, then log in. Now the function of creating a new application will be available to us. At first, we give it a name, then all links to it, the public key, and the application ID will be available to us. We can also change the previous name, logo, or description. On the left side as it is shown in Fig. 7, there is also a menu for creating a bot, and for specifying app testers.

But first, we set all the necessary permissions. We indicate that we are creating exactly the bot that will be used inside the application, and we can give it administrator rights. We also have access to many other functions, such as “View audit log”, server management, role management, sending messages, creating invitations, changing nicknames, and many others. But administrator rights as shown in Fig. 8, cover all these functions, and that is why we use this functionality. Save the changes, and click on the button to create a bot.

A chatbot can have many roles and functions as a salesperson on your website, provide sales services, generate invoices, pay for an order, or track its status, it can be seen in Fig. 4. The bot can also integrate with the corresponding APIs to fulfill client requests, provide them with authorization, integration with mobile or web services, and after integration, issue the result in the desired format. The bot can also act as an information provider or helper, provide automation of processes on the website, for example, demonstrate the rules, introduce new participants, or advertise something.

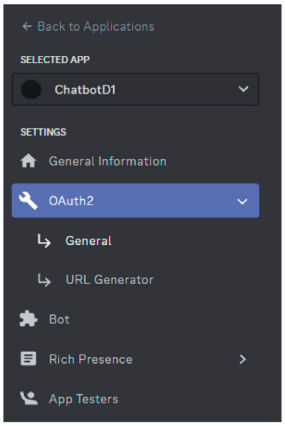

Figure 7 – Discord Developer Portal (Application Section)

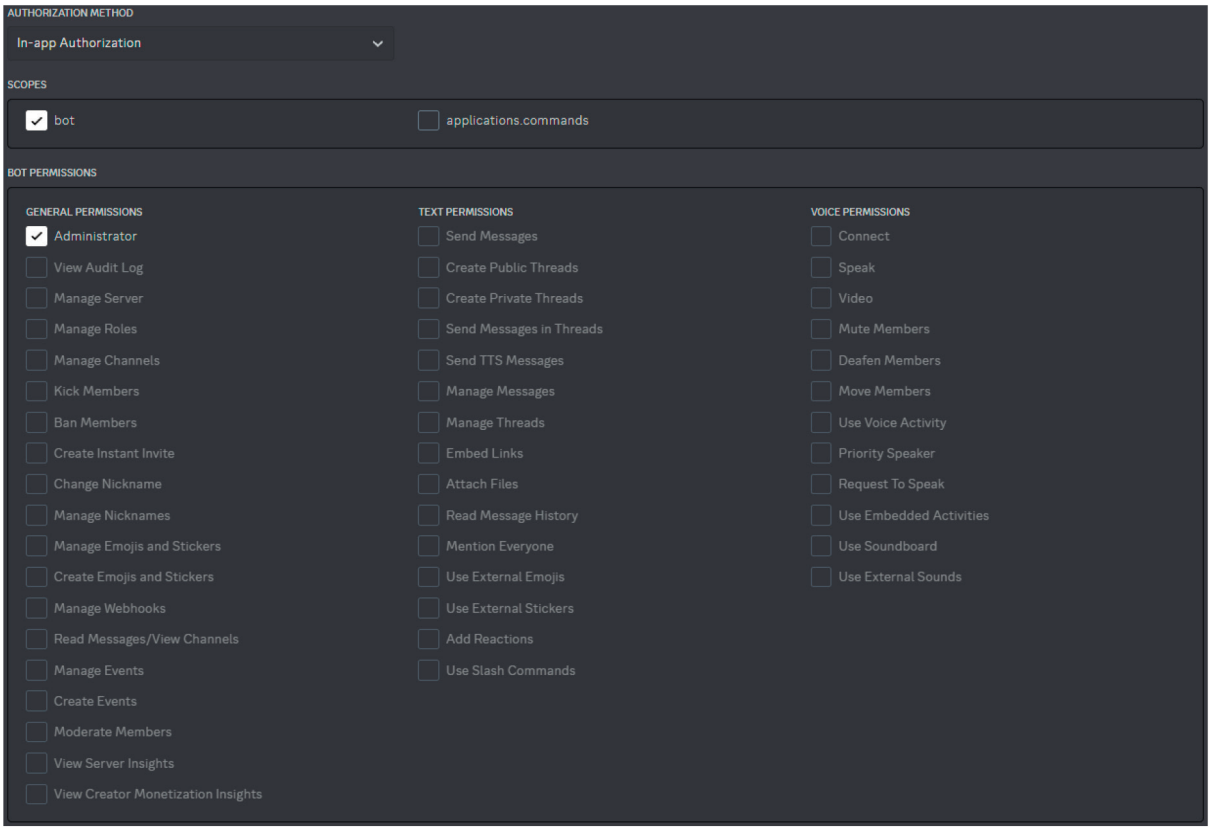

Figure 8 – Bot Permissions

Now we have access to links to invite the bot to the server, as well as the token of the bot itself, which will later be used in the code and will be responsible for direct integration with the bot. Therefore, do not lose access to the token, but in any case, it can be changed at the time you need it.

Our bot has been created, but now we need to work with the code, create all the necessary functions, and for this, Python will be used next.

The first step is to create a main file, in our case, it was named chatbotd1.py. Here the code was written to launch the chatbot itself using its token, the libraries of discord, itertools, os, asyncio, json, youtube-dl, pytube were also imported. The main code also contains many functions responsible for the state of the files, the status of the bot, and the creation and opening of files that will have their own code for individual bot functions.

All bot functions work with the help of special commands, these commands are set using a prefix. In our case, default prefix is a '!' symbol, for example "!setprefix =", the command to change the prefix from '!' to '='. This is done so that people can change prefixes to suit their needs. There are many bots, there are also many people, and in order not to get confused and use the right bot, people can change prefixes. Also, one bot can be added for several servers. In such cases, the prefix is written in the prefixes.json file. This file contains a prefix that is set by the owner of the server and the ID of the server itself, so the bot could work on different servers using different prefixes. When the bot is removed from the server, the prefix data with the ID of the corresponding server is also removed from the JSON file.

As we can see in Fig. 9, our code is divided into two main parts. This is the main code and cog files, that is, additional files that contain code for individual chatbot functions. When we run the code, there are messages that each of these modules runs correctly, so we can find out where we got the error. JSON files are also stored inside cog files, roughly speaking our database.

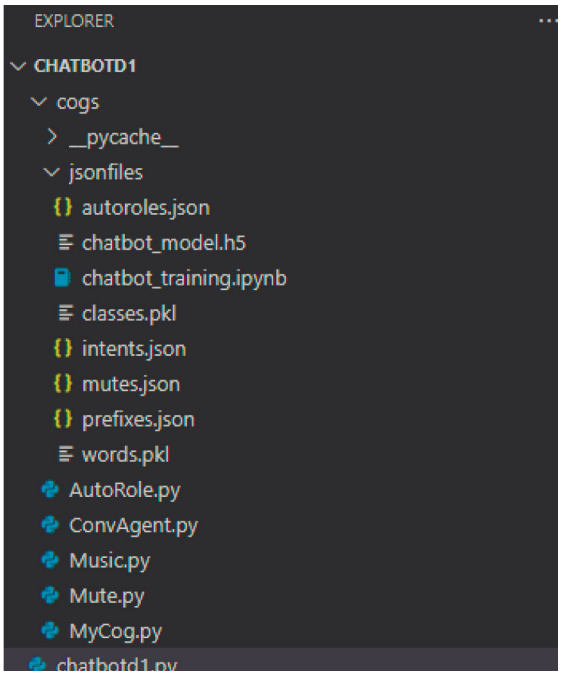

Figure 9 – ChatbotD1 Explorer

The chatbot profile looks as shown in Fig. 10, there is a button to add to the server, and code has also been written to show the status of the bot, which works cyclically. The date of its creation and the date of its addition to your server are also indicated.

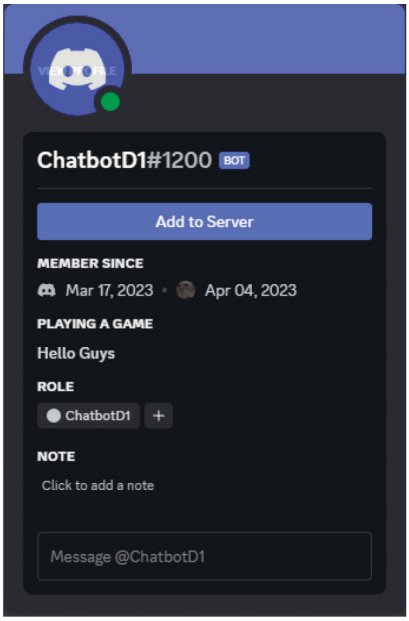

Figure 10 – ChatbotD1 Profile

After creating the main file, the next step was to create a cog file called MyCog.py. Functions such as Ping, Clear, Kick, Ban, and Unban were written here. These are basic functions, the initial task of which was to check the performance of the bot. The ping function provides us with data on the response delay from the bot, roughly speaking, the time that the bot spends on a response. Using the 'clear' function, you can delete the messages, the 'kick' function may be required when a user or bot needs to be excluded from the server. The 'ban' is responsible for ensuring that unfriendly users are not just kicked out, but also added to the blacklist of this server so that the user cannot connect to the server on his own. Well, if there is some kind of error, you can always pull this user from the blacklist using the 'unban' function.

The next cog file, called AutoRole.py, is responsible for the roles that are issued when a new user joins the server. For example, all your friends will have the role 'friend', and a new user joining your server automatically receives the role 'passerby'. Such data is also stored in a JSON file named autoroles.json, where the server id and the name of the auto role will be indicated.

The next function that has been added to the bot is called 'mute'. It is needed so that a person with this role cannot use voice chat. For this feature to work, we first create a role that will not have access to voice chat and define it as a mute role. This data is also saved in a JSON file and contains the server ID, as well as the name of the mute role. Now, using the simple command "!mute <ID>", you can assign this role to the user, and you can also remove this role with the "!unmute <ID>" command.

The following function 'play' allows the bot to play music in voice chat. Thus, by joining a voice channel, you can listen to music, a podcast, an audio movie, or something else with your friends. It is enough to write the '!play <song name>' command and the name of the song you need, the bot will download the most suitable audio and play it in the voice chat. This function is directly related to the YouTube API, namely the 'YouTube_DL' library, the search is based on the most appropriate result. One of the important criteria is that the user must be in any voice channel of the server. After the user requests any audio, it is added to the queue, then played, all subsequent requests will also be added to the queue and will be played in turn. All songs are downloaded to a temporary file called 'song', and stored there in binary form. There are also functions to pause, stop the audio completely, and play the next audio. An important part of this bot function is the 'ffmpeg' library, which is responsible for playing audio in voice chat. Even if the chatbot can download music, without this library, the chatbot will not be able to play music in the voice chat. In order to hear all the audio, the first thing you need to do is download this library and add the repository of this library to the path of the environment variable.

Inside the ConvAgent.py file, a neural network model is used, which we trained in advance. With the help of machine learning, the chatbot has become even smarter, now it can answer some questions from users, and we gave him the name, Goldy. The chatbot is based on a model that was trained on a database that we named "intents.json". For the chatbot to answer any question, we must refer to it using the "!g <question>" command. If the chatbot does not give you the correct answer, you can leave your feedback, they will be taken into account and the database will be expanded as far as possible.

5 Results

Let's see how our chatbot works and what it can provide us with. First of all, we can check the 'play' function, we will write the command "!play Pittbull", while we need to be in the voice channel, otherwise there will be a playback error as in Fig. 11. and the chatbot will send us a message.

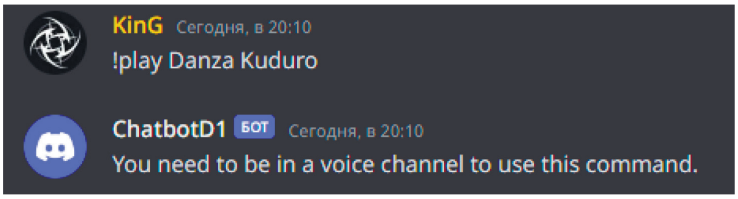

Figure 11 – 'Voice Channel' Message

When a user is in a voice channel, they can request a song, and further song requests will be added to the queue. You can also use the 'next' command to switch to another song and use the 'stop' command to stop the playback of a song. If there is no song in the queue and the chatbot is not currently playing the song, when calling the 'next' and 'stop' commands, the chatbot will display the following message "No song is currently playing". Fig. 12 shows how these requests were made.

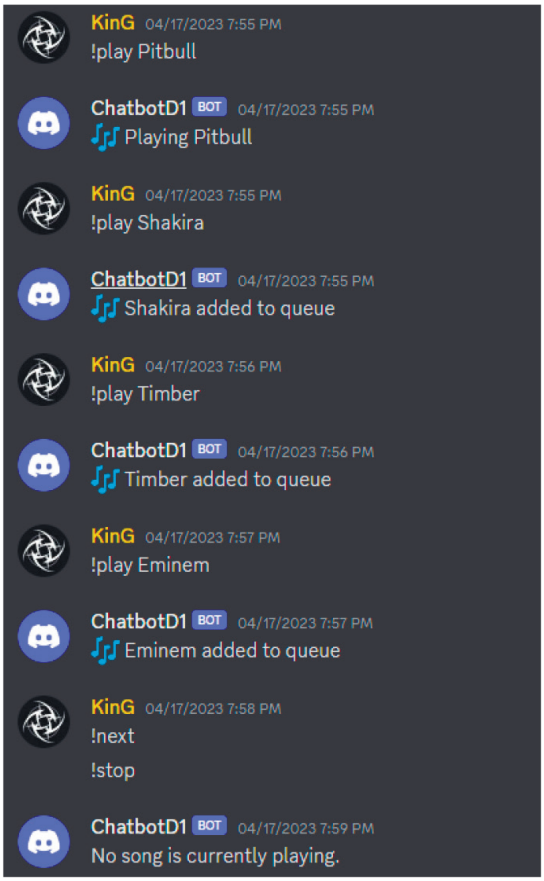

Figure 12 – Songs Queue

Now let's see how the 'ping', 'joinrole', and 'setmuterole' functions work. The ping function is quite simple, it calculates the delay from the client side and shows it in milliseconds. For the 'joinrole' and 'setmuterole' functions to work, we first select the appropriate roles. For the role that is automatically set, the role of the 'Universal Soldier' was selected, and for the 'setmuterole', respectively, 'muted'. And to set them, we need to enter a command and specify the role ID we need. To see the IDs of the required roles, you need to enable the developer mode in the discord settings. After we enter the commands, we can see that the chatbot sends us a message that the roles have been successfully set. This is done with a special feature called Embed. Now, when a new member enters our server, he will automatically receive the role of 'Universal Soldier', and unwanted guests can be muted using the mute command and they will get the appropriate role. In Fig. 13 you can see how we set these roles.

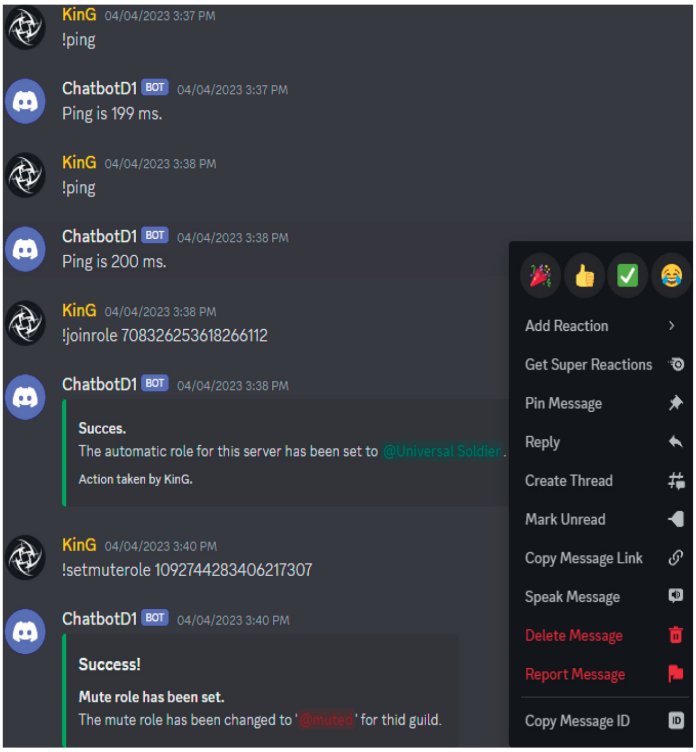

Figure 13 – Set Roles (Auto and mute roles)

The following functions use user IDs for work, these are the kick, ban and unban functions. We write the command "!kick <User ID>" and thus indicate which command to apply and for which user, respectively.

If, when using one of these functions, we do not specify the user ID, then the chatbot will display the following message "Error: Missing Required Argument. User ID or @ mention is required to kick". This type of error handling is also used for other functions, such as the clear function, auto role, and mute. Upon successful execution of the command, using the embed feature, we will see a message that will show us who was kicked, banned, or unbanned, as well as who performed this action and the reason for this action. Fig.14 shows an illustrative example of the commands with the user ID.

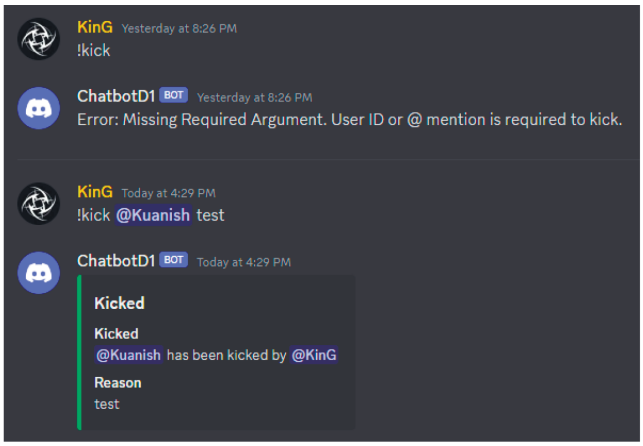

Figure 14 – Commands with ID (Kick, Ban, Unban)

The next feature of the chatbot is its ability to conduct a dialogue with the user. It is based on machine learning, and you can ask your questions and start a dialogue using the "!g <Your Question>" command. An example of a dialogue with a chatbot can be seen in Fig. 15. Here we learned that the chatbot's name is Goldy and that he can tell jokes. But these are not all the questions that the chatbot can answer, the database contains many more questions that you can get an answer to.

Figure 15 – Conversation with Chatbot

For the chatbot to answer our questions, a machine-learning model was created and trained on a dataset that was defined in a JSON file. This simple database is divided into three main parts, the first part is our tags, in other words, the general name of the questions that the user can ask. The next part is the questions that users ask, based on these questions, our chatbot is trained, and it gives the answer based on the third and last part of the database, these are the responses. Answers to user questions are written here, for example, to a request to tell a joke, the chatbot can give out more than 150 different types of jokes, which already adds some uniqueness to our chatbot. Fig.16 shows how the intents.json file looks like from the inside.

Figure 16 – Intents of chatbot

These intents and patterns were tokenized and preprocessed using the NLTK library in Python. Lemmatization, stemming, and removal of unnecessary characters like question marks and commas were also involved in the preprocessing. The resulting words were then sorted and saved as a set.

Next, classes were created based on our tags in the JSON file, as we said earlier, this is the common name for our similar questions. The data for training our model was selected by looping through each pattern in each of the intents and creating a bag of words for these patterns. A bag of words is a vector that represents the occurrence of each word in a pattern. It has been added to the output row corresponding to the tag, and the output row is then appended to the training data. The training data was then shuffled and split into the input (bag of words) and output (tag) arrays, or in other words our train_x and train_y.

A neural network model was then created using the Keras API of TensorFlow. The model consisted of three layers – an input layer with the same number of nodes as the length of the input array, a hidden layer with 128 nodes and a rectified linear unit (ReLU) activation function, and an output layer with the same number of nodes as the length of the output array and a softmax activation function. The model was compiled with a stochastic gradient descent (SGD) optimizer and a categorical cross-entropy loss function. It was trained on the input and output arrays using a batch size of 5 and an epoch count of 200. Finally, the trained model was saved as an H5 file using the 'save()' method of the Keras API. The resulting model can be used to predict the tag of new user inputs and generate appropriate responses.

Initially, to conduct a dialogue with users, a pre-trained model from OpenAI on the Curie engine was used, similar to ChatGPT. But since the bot gave answers that were not related to the question as it is shown in Fig. 17, it was decided to create a new model, with an intents-based dataset.

For the chatbot to work correctly, three different models were initially created. These models differed in optimizers, the first model had an SGD optimizer, the second Adam, and the third RMSprop. They all had 200 epochs and a batch size of 5. As we can see in Table 2, all three models have high training accuracies, but the testing accuracies are different and have a lower value, which may mean some overfitting. The first model with the SGD optimizer has the largest accuracy, then the model with the Adam optimizer and RMSprop follow in turn. The first model has the fastest training time, while the third model with the RMSprop optimizer has the slowest. The inference time is slightly different.

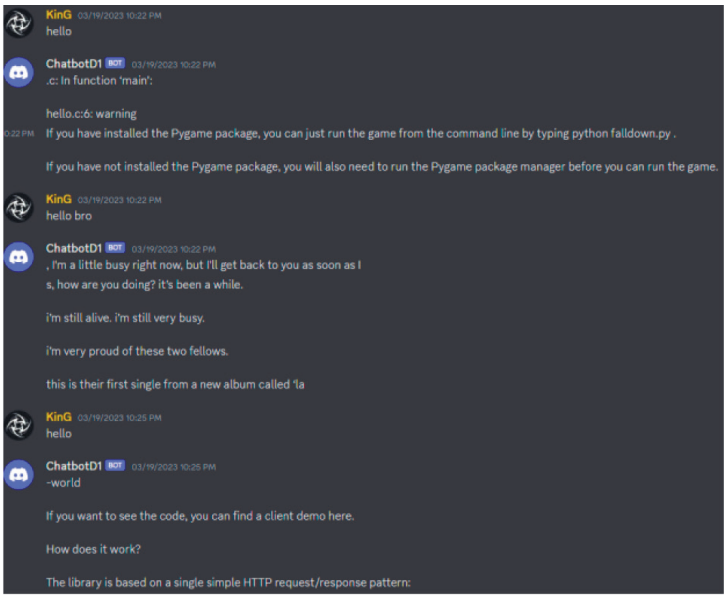

Figure 17 – Answers from Curie engine based chatbot.

Table 2 – Model Comparison

Model Optimizer	Training Accuracy	Testing Accuracy	Training Time (s)	Inference Time (ms)
SGD	0.98	0.87	23.05	1.16
Adam	0.96	0.85	38.15	1.23
RMSprop	0.95	0.93	54.78	1.34

Overall, the first model seems to be the most optimal and efficient as it has the fastest training time and the highest training and testing accuracies, therefore, this model, with the SGD optimizer, became the basis for the chatbot.

Discussion

Chatbots have become well-developed nowadays. They are gradually being introduced into all contact systems and allow them to save not only time but also their budget. According to a Juniper Research report from chatbotsjournal.com [14], the Chatbots Market was worth USD 1274.428 million in 2018 and is projected to reach USD 7591.82 million by 2024 registering a Compound Annual Growth Rate (CAGR) of 34.75% over the period (2019–2024). Fig. 18 shows trends from 2018 to 2023 for chatbot keyword. It can be seen that in 2022 the popularity of chatbots has grown dramatically, and this is due to the entry into the open market of the ChatGPT [15] chatbot from OpenAI.

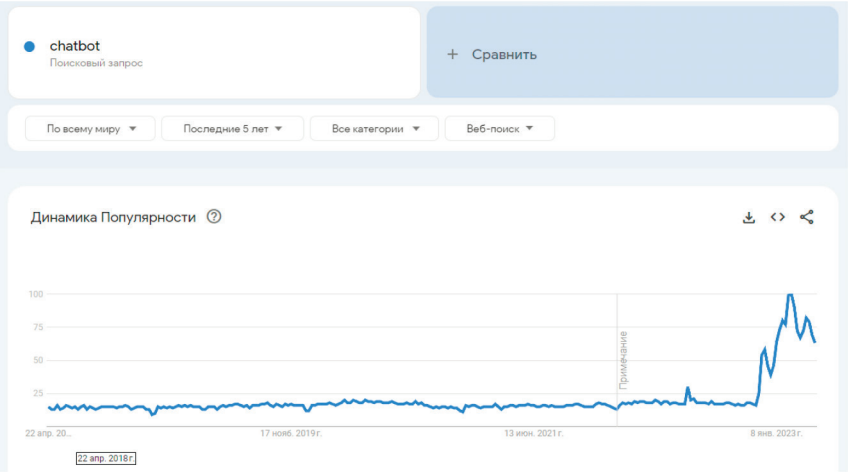

Figure 18 – Google Trends: Chatbots (from 2018 until 2023)

Chatbots have already been introduced to such top business representatives as Bank of America, Visa, Mastercard, American Express, Barclays, and PayPal.

What benefits have chatbots been able to bring to such giants?

Lower Operational Costs: According to statistics from chatbotsjournal.com, chatbots will save 11 billion USD by 2023. Such saves will come true as chatbots are getting smarter every day.

They help users with their simplest and most frequent questions, which makes the contact center assistance process more efficient.

Chatbots were able to replace not only voice calls and responses, but also reduced the number of messages in business mail and social networks. Because chatbots are communicated with before contacting company representatives.

Reduced Labor Expenses: Organizations using chatbots as their first line of aid have been able to achieve significant savings in labor costs. Since one person can simultaneously communicate with a maximum of three people, chatbots can serve an unlimited number of people. Chatbots can be used to solve simple problems, and a human agent can solve complex queries as needed.

24/7 availability and low cost: Chatbots can work for 24 hours every day while serving customers [16]. So chatbots allow ordinary workers not to go out on night shifts and can work all year round, giving answers to all customers.

Continuous Savings in the Future: As we know, many chatbots use machine learning and can improve every day. With their development, calls from customers will also become less frequent and chatbots will become smarter, which means that in the future, their effectiveness will not dry out and will be at its peak.

Conclusion

In conclusion, we can say that chatbots are tools that significantly change the entire system of the implemented area. Areas such as medicine, sports, entertainment, education, or the economy. They significantly improve the quality of user service and help to find the information they need very quickly, without wasting their nerves, money, and time.

Chatbots are assistants who give you answers using Artificial Intelligence and NLP, and who can help you in your daily life with any task, from simply finding the music you want, to developing the initial structure of a website.

A functioning chatbot was developed for the Discord platform, the main feature of which is the ability to conduct a dialogue with the user. The basis for such a chatbot was the neural network model, with the SGD optimizer. In addition, the chatbot has some other features that allow you to use the capabilities of the Discord platform more effectively. In the future, it is also planned to work with other APIs except for YouTube, adding several different functions, as well as further expanding the database, which will allow the neural network model of the chatbot to conduct a dialogue even more efficiently.

It was concluded that the most effective chatbots are those that are used to give out any information, in the form of an assistant, guide, or friend. We learned that such chatbots can increase the revenues of companies and reduce their costs to a significant extent.

Due to the current pace of development in the field of artificial intelligence in general, we hope that in the future chatbots will be used even more actively and will gradually be introduced into all informational, entertainment, and other websites.

References

1. Abu Shawar, B., Atwell, E. (2007). “Chatbots: Are they Really Useful?” University of Leeds.

2. Adamopoulou, E., & Moussiades, L. (2020). “Chatbots: History, technology, and applications.” International Hellenic University.

3. Singh, A., Ansari, M., Shaikh, S., Parbulkar, M. S., & Khan, T. (2021). “Intelligent Chatbot.” Maharashtra Institute of Technology.

4. Behera, B. (2016). “Chappie – A Semi-automatic Intelligent Chatbot.” Vellore Institute of Technology.

5. Vakili, A., & Shakery, A. (2019). “Enriching Conversation Context in Retrieval-based Chatbots.” University of Tehran.

6. Følstad, A., & Brandtzaeg, P. B. (2017). “Why People Use Chatbots?” *SINTEF Digital*.

7. Thakkar, J., Raut, P., Doshi, Y., & Parekh, K. (2018). “Erasmus-AI Chatbot.” University Rotterdam.

8. Abraham, J. (2018). The Significance of Chatbots. Retrieved from *Dzone* Website: https://dzone.com

9. Hancock, B. (2019). “Learning from Dialogue after Deployment: Feed Yourself, Chatbot!” Computer Science Dept. Stanford University.

10. Pal, A., Bhalotia, S., Rathod, V., Bisen, S., & Lalwani, T. (2018). “Implementation of a Chatbot System using AI and NLP.” K. J. Somaiya Institute of Engineering and Information Technology.

11. Ghias, A. R., Guo, C., Sarikaya, R., & Ponnusamy, P. (2019). “Amazon Alexa, Feedback-Based Self-Learning in Large-Scale Conversational AI Agents.” Retrieved from website *Amazon*: https://www.amazon.com

12. Farhan, L., Kaiwartya, O., & Kharel, R. (2018). “A Concise Review on Internet of Things (IoT) – Problems, Challenges and Opportunities.” University of Oulu.

13. Caldarini, G., Jaf, S., & McGarry, K. (2021). “A Literature Survey of Recent Advances in Chatbots.”

14. Brain [Brn.Ai] Code for Equity (2021). “Chatbot Trends Report 2021.” Retrieved from *chatbotsjournal* website: https://chatbotsjournal.com/chatbot-trends-report-2021-b15479c404e4

15. Khosravi, H., Shafie, M. R., Hajiabadi, M., Raihan, A. S., & Ahmed, I. (2021). “Chatbots and ChatGPT: A bibliometric analysis and systematic review of publications in Web of Science and Scopus databases.” *Journal of Information Science*, 47(5), 550-573.

16. Benlian, A., Adam, M., & Wessel, M. (2020). “AI-based chatbots in customer service and their effects on user compliance.” Technical University of Munich.