International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 09 Issue: 04 | Apr 2022 www.irjet.net p-ISSN: 2395-0072

VOCAL- Voice Command Application using Artificial Intelligence

AKSHAY DNYANESHWAR GOVIND[1], SAURABH ASHOK JADHAV[2], DHRUV KUMAR GUJAR[3]

[1,2,3] *(Students of VPPCOEVA, Mumbai)*
[4] *Under Guidance of – Prof. VINOD ALONE Dept. Computer of Engineering, VPPCOEVA, Mumbai, India*

Abstract - *A chatbot is artificial intelligence (AI) computer software that can simulate a conversation using textual or audio techniques. The basis of chatbots is artificial intelligence, which analyses a customer's data and blends the answer with them. AI-powered bots can take over a variety of duties since they are considerably more powerful—and can execute numerous tasks at once. Machine Learning methods are mostly employed in the process of comprehending and responding to user input. Natural language processing enables a bot to converse in the most natural manner possible. A balanced blend of innovative technology and human intervention is the optimal user-chatbot connection.*

***Key Words*: NLU Engine, NLG Engine, ML Engine, Architecture, Chatbot, Chat Interface.**

1.INTRODUCTION

A chatbot may have intelligent conversations via text or speech. Chatbots are computer programmes that mimic user behaviour on one side of a talking conversation. They're mimic systems that pretend to be two people having a discussion. They serve as a model for successful and intelligent communication with the user on the other end. They work to give services similar to those provided by marketers, salespeople, counsellors, and other mediators. Business, market, stock, customer care, healthcare, counselling, recommendation systems, support system, entertainment, brokering, journalism, online food and accessory buying, travel chatbots, banking chatbots, recipe guides, and so on are just a few of the fields where chatbots may be found. The most well-known chatbots, such as Alexa and Google Assistant, are the greatest instances of smart conversing chatbots. These are all-purpose chatbots that offer services across all domains and are not limited to a single one. Domain-specific chatbots are also available, which give functionality to the aforementioned domains. Types of chatbots used:

Support – This is used to master a single domain. Skills - This does not require a lot of contextual awareness.

Assistant - This is the middle ground between a skill and support chatbot. When they know a little about a variety of topics they work great.

There are several existent chatbots for diverse domains that perform various functions. In the medical industry, there are several chatbots. The issue with these chatbots is that they just respond to users' inquiries in a boring manner. They are unable of establishing intelligent conversation with the user in the same way that a human can. The goal of the study is to provide a chatbot system that can build smart dialogue. This may be accomplished by building a clever and responsive chatbot that can converse with the user in the same way that a human can. NLU, NLG, NLP, and ML algorithms must be introduced into the system to make traditional chatbots behave like virtual buddies. These strategies make the system more conversational in plain language, are useful for counselling, and may be used to simulate illness prediction.

1.1 Previous Researches

Chatbots used for customer service in the telecom and marketing industries are scripted chatbots. They assist consumers with predetermined customer service queries. Traditional monotone chatbots are being studied in order to make them talkative, responsive, and capable of communicating in natural (conversational) language. This necessitates the use of NLP and machine learning algorithms in the system. There are several options for doing so. The domain of the chatbot, the capabilities it seeks to give, the communication language, the end user, and so on all influence the technique used. When creating a chatbot system, all of the aforementioned difficulties must be taken into account. Some of the methods are briefed about in the following section.

In the history of chatbots, there have been numerous notable achievements. A.L.I.C.E (Artificial Linguistic Internet Computer Entity) is a highly regarded programme that was used to create this free chatbot. A.L.I.C.E makes use of Python's AIML (Artificial Intelligence Mark-up Language) to provide solutions to our inquiries or inputs. A domain specific knowledge system is a collection of AIML files that make up a knowledge base system. AIML is a chatbot and human interaction language based on the XIML language. A collection of pattern and template is the most important aspect of AIML. Not just this chatbot, but all of the chatbots that have been constructed, have gotten a lot of interest throughout the world. However, they only live for a brief time.

The "Turing test" is one of the most commonly acknowledged measurements of a system's intelligence. According to the Turing test, if a group of individuals connecting with an unknown unit (for example, through

keyboard) thinks that the entity is human although the unit is really a mainframe, the mainframe is said to be accepted. Natural language processing (NLP) allows computers to communicate with one other using human natural language, allowing for user-to-computer or human-to-machine communication as well as computer-to-computer or machine-to-machine communication.

2. Literature Review

Chat Interface

The system's front end is this unit. It is in charge of gathering the user's inquiries, which serve as input to the system. It's also in charge of showing the user the results generated by the system. As a result, the chat interface might be considered the system's face, through which all conversation takes place. It serves as a conduit for communication between the system and the user. The chatting backend, which works as a message delivery mechanism between the Chat interface and the Machine Learning Layer, receives the user's inquiry on the chat interfaceThis interface can be accessed via a website, a smart phone app, or built-in software. The type of user interface is determined by the system's ability to meet the user's needs. If the system is accessible through a smartphone, the interface will be in the form of an app; if the system is accessed via a website, the interface will be in the form of a website; alternatively we may call it a system software that handles all of the computer or smartphone's functions. It will be necessary to utilise Android for Android phones or Swift for iOS to develop apps for smartphones. Only the interface platform will be developed in Android in this scenario, and the system's backend processing will be done on a server where the system will be deployed. Java or Python web frameworks can be used to create a website. The most powerful and up-to-date web frameworks in Java are Spring and Struts. Similarly, the Django and Flask frameworks in Python allow you to create a website. The system's intended functionality, the demands of the people who will use the system, the algorithms that will be used by the system, and so on all influence the criteria for selecting a programming language. Selecting the right programming language makes it easier for developers to create a system that gives maximum functionality to the user while maintaining high accuracy and simplicity.

NLU Engine

Natural Language Understanding (NLU) is a subset of Natural Language Processing (NLP) that allows a system to understand natural language or conversational language spoken by people. Conversational language, which is utilised by humans in everyday talks, is not as flawless as formal language. It does not place a strong emphasis on vocabulary or grammar. As a result, a system's ability to interpret the sentence's intent is hampered. The user's input is in an unstructured text format that the system cannot understand. It only accepts input in organised forms. The unstructured text obtained from the user is transformed to structured format by applying NLU algorithms to extract relevant terms and patterns from the user content. Mispronunciations, homophones, swapped words, shorter forms of words (such "it's" for "it is"), slang words or phrases, and terms not used in formal language but found in everyday discussions are all understandable to humans. NLU methods allow the system to recognise these twerks if they are used by the user while interacting with the chatbot, giving the impression that the discussion is between two humans rather than a person and a bot. The meaning of the user utterances is not immediately understood by NLU systems. It entails a series of steps to determine the sentence's true intent. Each word of a phrase must be understood by the NLU system in order for it to comprehend the entire sentence. This implies that the first step is to break down the phrases into individual words. After then, the machine must comprehend the sentence's syntax in order to comprehend the term. This may be accomplished by understanding the parts of speech associated with each word in the phrase. The POS (Parts-Of-Speech) tagger enters the scene. After determining the grammatical weightage of each word, the dependencies between them are determined. This is the most critical stage, in which the word with the highest dependence is retrieved and the system's goal is deduced. It's unlikely that the knowledge base will include the identical language submitted by the user. It might contain a phrase with the same aim as the previous one, but with different terms. Synonym determination and sentence matching are necessary to match these sorts of synonymic statements. The many tasks to be implemented under the NLU Engine, as well as the techniques to do so, have been addressed in further detail.

1. Word Segmentation

Segmentation, also known as tokenization, is the act of breaking down large amounts of text into smaller, more relevant chunks. Paragraphs, sentences, clauses, phrases, words, and letters are examples of these units. The letters are the smallest unit. The separating of sentences into individual words separated by blank spaces is known as word segmentation. Tokens refer to the tokenized units of the sentences. The tokenizers separated the sentences into individual words and punctuation marks. The most frequent tokenizer is of the space type, which divides sentences into words at blank spaces. It is also necessary for the tokenizer to take into account abbreviations, acronyms, dates, numerals in decimal forms, and other characters that cannot be separated at punctuations or blank spaces without losing their significance.

Mohammed Javed et al. described a technique for implementing word segmentation In his approach, he suggested calculating character spaces in sentences. All forms of gaps between characters should be included in the

character spaces. Gaps between letters, punctuation, and words are among them. The gap or character space between each unit in the text determines how the algorithm works. To get the mean average between characters in the sentence, an average of the gaps is determined after the character spaces are calculated. After that, the average gap distance is applied to the sentence to be split. The sites of tokenization are the locations where the character space exceeds the average character space. The gap between words is always more than the average gap and hence tokenization takes place at the blank spaces between words in the sentences.

Naeun Lee et al. advocated the use of NLTK to achieve word segmentation. Natural Language ToolKit (NLTK) is a Python module that provides services for Natural Language Processing (NLP). It has tokenizers built in. Users must first import the package and then use the appropriate tokenizer, which is available in the form of functions. Standard, letter, word, classic, lowercase, N-gram, pattern, keyword, path, and others are among the tokenizers included in the NLTK. The word-punkt tokenizer, which breaks sentences at blank spaces, is the most often used tokenizer. The NLTK tokenizers are impressive in terms of accuracy, speed, and efficiency. It also doesn't necessitate any algorithm implementation because the package runs them in the background.

For word segmentation, Tao Jaing describes how to use the CRF (Conditional Random Fields) Algorithm. This method prepares the computer to recognize gaps between letters. The algorithm recognizes the space between characters in the test sentence using this training. For the gap distance, the system maintains a threshold value. If the value of gaps in the test sentence is more than the specified threshold, then the sentence splits at those points CRF necessitates a significant amount of training for the system, which prolongs the process. When comparing the three ways shown above, the NLTK proves to be more efficient in every way than the other two. The use of NLTK does not necessitate the implementation of any algorithm because the package handles everything. Furthermore, the package's accuracy, speed, and diversity outperform the two methods.

2. POS tagging

The practice of giving grammatical annotations to specific words in sentences is known as POS Tagging. The Parts-Of-Speech Tags are among the annotations. They indicate the grammatical importance of a word in a sentence based on the word's reliance on other words in that phrase, clause, sentence, paragraph, and so on. Noun, verb, pronoun, and other POS tags are widespread. There are a variety of approaches that may be utilized to accomplish POS Tagging. Some of these are discussed farther down.

Jerome R. Bellegarda for POS Tagging, an approach termed latent analogy was presented. The latent semantic mapping (LSM) approach is employed in this algorithm. It necessitates training on the corpus supplied. The LSM keeps track of the tagged feature space of the training corpus. Now, new phrases are fed into the LSM for tagging, and the analysis is run to see which sentences from the training data are the most similar to the test sentence. Sentence neighbourhood is the term for this. If two sentences have the same intent matter, sentence neighbourhood holds true. Following the discovery of intent matching sentences in the training data, the POS tags associated with those sentences are mapped to the test sentences.

Liner Yang et al. presented a method for implementing the POS Tagger with Neural Networks. This method has a total of "n" hidden layers. The amount of iterations or combinations necessary to appropriately tag the relevant sentence determines these layers. Each word in the phrase is tagged with an appropriate POS tag at each layer of the algorithm, and the tags are then transmitted to the next layer for verification. Unless the following layer gives the same tags as the preceding layer, this will continue. Another method for implementing the POS tagger is to use the standard approach, which is to keep a dictionary of tags for the language in question. The Python NLTK package has a built-in Tagger that may be used by just importing the NLTK module. The NLTK comes with a preset set of tags as well as its own training data. It evaluates the statement and assigns the appropriate tag. When comparing the three methods above, the NLTK tagger comes out on top in terms of speed and efficiency. The neural network approach, on the other hand, provides the maximum accuracy because it goes through numerous iterations.

3. Dependency Parsing

A dependency parser is a programme that uses grammatical tags to determine the link between words in a phrase. After parsing, it's time to move on to the following stage. For each statement, a dependency tree or graph is built. The dependency tree or parsing tree is the name given to this tree. There are several approaches to parsing that may be used. The following is a comparison between the two.

Bo Chen A technique for implementing the dependency tree has been presented. It begins by determining the interdependencies between the terms in the phrase. Each word is examined to see whether it has a relationship or is dependent on the other. The root is chosen as the term with the most dependencies. The child nodes are the other words that have a relationship with the root node. This process continues until all of the words have been included into the tree. The dependency parser tree is the tree form of the sentence. Using the POS tags, the dependencies between the words are discovered.

Zhenghua Li presented a more adapted dependency parser model the parser constructs a parsed tree for the needed sentence in the typical approach described above. The tree is transformed to a graph in the graph-based dependency

parser, where the words in the sentences are the vertices and the dependence between the words is represented by the edges. The parsed sentence is best represented in this data format. The usual technique of parsing must always be used. However, a graph-based parser increases the parser's visibility, readability, and comprehension.

4. Synonym and Pattern Recognition

No sentence given by the user can be absolutely identical to any sentence in the database, regardless of how large our data is. However, there are phrases that have the same meaning. After determining the user's purpose, the database is searched for a statement with the same meaning. The words chosen to represent the same information in the matched phrases are different. They substitute words or use synonyms. As a result, synonym detection is required by the system. Domain independent or domain dependent synonyms exist for a given term. Synonyms for a term throughout the full lexicon are known as domain independent synonyms. Domain-dependent synonyms, on the other hand, are synonyms for a single term in that domain. There are various algorithms used for the detection and extraction of synonyms, some of which are reviewed below.

LinHua Gao et al. describes how to extract synonyms using the classic dictionary technique. The system database maintains a collection of synonyms for relevant terms in that area using this way. After then, the user's statement is mapped to the synonym dataset. The discovered keywords from the sentence are then compared to the synonym list to see if they have the same intent. After that, all possible synonyms of that term are searched in the main database for a match. The closest sentence to the user sentence is retrieved. This approach takes longer and necessitates greater storage and complexity.

Sijun Qin For synonym extraction, a feature selection strategy was presented. Positive tags for nouns, verbs, and adjectives are indicated as positive tags in this manner, whereas negative tags are marked for the rest of the parts of speech tags. Using the POS tags, the polarity for each feature (word) is then determined. It may be definitely determined if the overall feature polarity is positive. As a whole sentence is tested for its synonymic interpretation, the synonyms detection for the collection of characteristics will be pretty strong. The information gain is then computed using the extracted synonym sets for that clause of characteristics. The one with the highest information gain is the strongest synonym extracted.

Decision on ML Engine

Scripted or repetitive chatbots have pre-programmed responses. They respond to the user by selecting from a list of predetermined responses grouped according to the user's question. The use of machine learning in chatbots allows them to generate responses from scratch. It's utilized to develop predictions about how users would respond to their inquiries, as well as to improve the system based on past experiences. It constantly updating the databases as new information from the user comes in. This engine analyses what the user demands using supervised, unsupervised, or both methodologies. It also employs a model to read the user's intent and deliver the right results. The outcomes might take the shape of forecasts or any other type of analysis based on the execution and analysis of mathematical models. The majority of machine learning models are built on statistical and probabilistic analyses of the instances that occur, with the computations resulting in a prediction for the test case. The decision engine incorporates not only prediction models, but also information retrieval methods such as entity extractions, multiple text classifications, and so on. A machine learning layer in a chatbot system is also used to construct an ontological link for entities retrieved, as well as to associate them with context-specific inquiries, as well as their alternatives, synonyms, and machine-enabled classes. These machine learning technologies transform a simple and static FAQ system into a smart and tailored communication experience. The machine learning layer extends the capabilities of chatbots that deliver services across a variety of disciplines. Its goal is to improve the accuracy of the system's replies to users while simultaneously broadening its breadth. By learning from its experiences, the system is able to update itself. As a result, the system is less likely to produce incorrect predictions. For illness prediction, chatbots in the healthcare area can utilize a variety of methods, including clustering, Bayesian networks, decision trees, and so on. The techniques of their execution, as well as a comparison of the algorithms for proper selection, are briefly discussed here. The system's brain is called a decision engine. It entails using machine learning algorithms to make predictions, as well as statistical and probabilistic computations. ML also allows the system to learn from its previous experiences, resulting in improved and updated outputs. Disease prediction algorithms are required for chatbots in the health-care area. Prediction may be done in a variety of methods, some of which are discussed here.

Sachin S. Gavankar et al. For prediction, the eager decision tree method was devised. The improvised variant of the classic decision tree is this form of decision tree. This tree is created at runtime depending on the user's queries, and it is updated when new user messages arrive. Consider how well it predicts illness. The symptoms observed in the user query are added to the root node as child nodes in this approach. New nodes are added as new symptoms are discovered. Furthermore, for each symptom, the algorithm looks for the second symptom that occurs most frequently with the first and asks the user for that symptom. If he says yes, then the system traces that path to check for the disease present at the root node. This will keep iterating for all users and the tree keeps getting updated for new entries or traces the path available.

Naganna Chetty et al. propose a hazy approach to forecasting the system follows the clustering technique in this method. That is, the algorithm takes facts from the knowledge base that is closest to the user query. When a user submits a query, the algorithm searches the knowledge base for the best matches and returns them to the user. The best results are further searched for relevance in the following iteration, when the user enters the second query. Every iteration filters the matches based on the user's query. This procedure is repeated until just one best match is found, at which point the user is given with a predicted result. When we compare the two techniques, we can see that fuzzy logic (clustering) prediction is easier to construct and has less complexity. The eager decision tree method, on the other hand, is more sophisticated and takes longer to execute. However, as compared to the fuzzy technique, eager decision trees yield better accuracy.

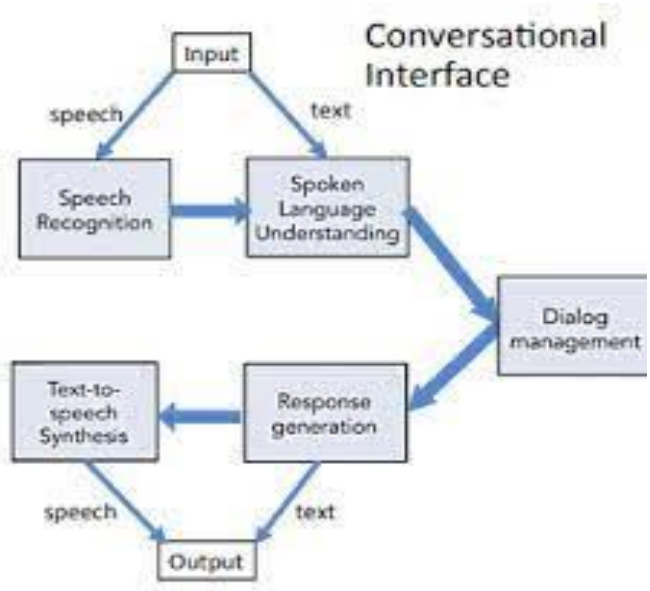
NLG Engine

NLG accomplishes the opposite of NLU. It is the process of translating the system's output into plain language representations that the user can understand. To put it another way, NLG is the process of creating text/speech from patterns created by a computer system. The system's output is in an organized manner, allowing it to be simply comprehended and processed by the system. NLG is a natural or conversational language representation of the system knowledge base that is easily understood by the user. A single statement can be expressed in a variety of different ways. The active or passive voice might be used in the sentences. A resemblance between two phrases can also exist, however it is possible that they will use synonyms. As a result, while presenting a response to the user, the NLG unit must first analyze all possible interpretations of the same statement before selecting the most relevant one. To create phrases, the NLG engine conducts a series of activities. The first step is to decide on the content. It entails deciding the response to offer to the user. This stage determines what kind of material (or combination of words) should be included in the sentence. It also takes into account the location of words in phrases based on their POS Tag (placements of verbs, nouns, adjectives, prepositions, etc.). Overall, this stage is concerned with the structure of a basic phrase, starting with the selection of words and ending with their placement in the sentence. The selection of sentences is the next job. As previously stated, there are a range of phrases that may be used to convey the same scenario; this stage is concerned with selecting the most appropriate sentence for that situation. The sentences considered for possibilities are in an abstract structure and are not whole sentences. To make them grammatically accurate, they must be supplemented with grammar rules. This part examines the semantic accuracy of statements using the system's grammar rules. The morphological check is the final and most critical phase, in which the phrase created in the preceding processes is evaluated for validity. This stage verifies that the sentence is correct.

3. Literature Survey on Various Chatbots

A."Task-based Interaction Chatbot", EEE521 final year project Report school of computing, Engineering & Intelligent System[4]. Authors: Dr. Kevin Curran, Dr. Daniel Kelly

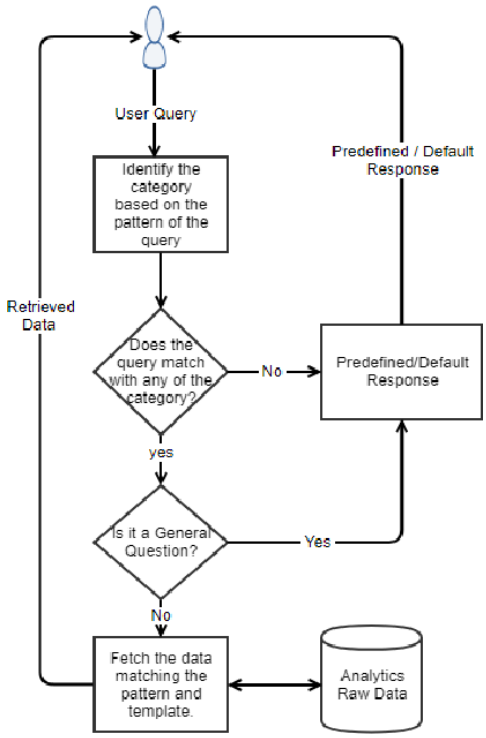
Chatbot Architecture consists of four sections Front-end comes first, followed by knowledge-based, back-end, and corpus (training data). The front end component is where the user is communicated with. Natural language understanding (NLU) is a technique for deciphering the context and intent of user input. A suitable answer is elicited from the user. The chatbot's knowledge is determined by the knowledge base, which is done using the NLU and supported at the back-end. The domain corpus is used by the backend to create the knowledge base. The chatbot receives input in the form of speech or text. The data is sent into the dialogue management system, which generates a suitable answer and instructs the chatbots to take the right action. The responses are produced in the form of text and speech both.



B. "Intelligent Chatbot for Easy Web-Analytics Insights". In 2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI) (pp. 2193-2195). Author: Ravi, R.

In this paper, a comparison is done based on their ease of usage, using different analytic tools. The chatbot is built using Artificial Intelligence Markup Language contain analytics' raw data and the required data is fetched from the analytics tool's raw data. Every website keeps all of the data that the user enters. AIML consists of queries and responses to those requests. There are three parts to it: a template, categories, and a pattern. Each category includes a pattern and a template. Patterns are the various inquiries that the bot-user can enter, and the template is the response to each pattern [5]. There are three query possibilities to consider [5]. Domain-Related Query (Scenario 1) Scenario 2:

Frequently Asked Questions Scenario 3: Users can type their question into web analytics and get a rapid answer to none of the above. Web analytics tools have been mastered to avoid the time-consuming task. Raw analytics data was used to create the system.



Vibhor Sharma et al. proposed a paper: "An Intelligent Behaviour Shown by Chatbot System"

This study compares and contrasts the two most popular chatbot systems, ELIZA and ALICE, as well as their uses. It explains how to use a domain-specific information system to provide answers to frequently asked questions in a university setting.

Sumit Wailthare et al. proposed a paper: "Artificial Intelligence Based Chat-Bot"

This paper presents a method for implementing a web-based AI chatbot as a personal assistant for users, which uses a pattern matching algorithm to speed up the arranging and starting of meetings between users and customers. The system's high efficiency makes it capable of being deployed in the real world.

Kedar S Chikorde et al. proposed a paper: "Natural Language Processing Based Chatbot For IoT Dashboards"

This paper discusses open-source programmes for building chatbots, such as Apache's OpenNLP. There are useful frameworks available, such as Google's API. AI, Amazon's Alexa, and a slew of other Internet-based tools that can be integrated directly into the software.

Balbir Singh Bani et al. proposed a paper: "College Enquiry Chatbot Using A.L.I.C.E"

This is an example of a successful chatbot. AIML is the software used by A.L.I.C.E. A.L.I.C.E appears to be a real person, but we must remember that we are conversing with a machine. It is mostly based on natural language comprehension and pattern matching. AIML is a simple XML language that aids humans in the creation of simple programmes. A.L.I.C.E accomplishes functions such as simplifying the input and combining two replies in the splitting, which can be caused by the Normalization or Recursive processes. A.L.I.C.E seeks to generate diverse answers in the situation of comparable inputs. Using grammatical analysis, it has the capacity to provide the derivation structure for a sentence.

Sameera A et al. proposed a paper: "Survey on Chatbot Design Techniques in Speech Conversation Systems"

This is primarily concerned with speech. The goal of research has been to improve human voice recognition rates, and the technology is now close to becoming viable for speech-based human-computer interaction. Voice interaction can be divided into several categories, such as speech recognition, speech passing, natural language processing, keyword detection, and so on. It is also stated that if speech recognition is universally recognize ed as the future interface, there will be no need for a mouse or keyboard. NLTK (National Language Toolkit) is a significant tool for dealing with voice recognition. In the field of speech communication systems, numerous strategies have been listed out to create chatbots. As a result, this research article delves deeply into these issues.

Hye-Kyung Cho et al. proposed a paper: "A Chatbot-Based Robot Control Architecture for Conversational Human Robots Interactions"

This research is primarily concerned with chatbots that have robotic interactions. There are a few disadvantages of using a chatbot to provide robotic services. Chatbots are typically designed to respond, but not to meet all of the requirements set forth by users. Following that, chatbots provide an explanation of the request-reaction pair. Robots are unable to function without the assistance of humans. The problems of a chatbot can be mitigated by designing and implementing a robot software architecture. We should create a robot that can manage many contexts at the same time. Robots should be designed in such a way that they can imitate human face expressions. This is how you create a chatbot-based architecture.

Ulrich Gnewuch et al. proposed a paper: "Faster is Not Always Better: Understanding the Effect of Dynamic Response Delays in Human-Chatbot Interaction"

The main focus of this research is on how robots and humans react to text message delays. People who use

chatbots perceive them as more human-like, socially present, and happy with the entire discussion when responses are dynamically delayed, rather than when they are supplied practically instantly. The amount of time it takes for a chatbot to respond is a social cue that causes users to react socially. Dynamic reaction delay is required when there is a situation with unique characteristics. Our findings represent an important first step in making chatbot-human interaction more natural.

M.Dahiya et al. proposed a paper: "A Tool of Conversation"

The design and implementation of a chatbot system are the topics of this study. The question of how intelligently a chatbot can connect with people is being debated, but only on the basis of a text-only chatbot. When asked a question, the chat bit will react in a predetermined pattern. A chatbot is primarily developed using pattern matching, in which the sequence of the sentence is identified and the user is provided a saved response pattern. It can be stated simply as follows: user -> ask question -> chatbot -> respond -> user. Factors to consider when building a chatbot include: 1) Operating System (OS) 2) Software Selection 3) Using a Chatbot to Cheat 4) Establishing a Chat 5) Matching Patterns 6) Easy to understand 7) Environmentally friendly and entertaining.
The following steps are included in the implementation process:
1) creating a dialogue box.
2) creating a database .
3) Modules Explanation.

4. Advantages and Disadvantages

Advantages

- Availability 24 hours a day, 7 days a week — Once installed, the chatbot can respond to requests at any time. Companies will be able to contact them afterwards as they track their activity outside of working hours.
- Self-updating and learning — Chatbots have the ability to update and learn from transactions on their own. They can update themselves thanks to algorithms and machine learning.
- Chatbots save money in the long run - Companies with human customer service teams have to pay a lot of money in salary to their personnel. And if the company is big, the expenses will be big, too! A chatbot can help the organisation save money in the long run by reducing this expense. While it is true that establishing a chatbot takes a lot of time and money at first, the chatbot will eventually be able to answer all of the fundamental client questions that would take people a long time to respond. Furthermore, the chatbot may provide fast responses, which saves time. In business, time is money!!!!!

Disadvantages

- Complicated User Interface - Chatbots take a long time to figure out what the user wants.
- Inability to Understand - If an unsaved query is presented to a chatbot, it will not react effectively owing to fixed programmes. This causes client unhappiness and, in some cases, loss.
- Time-consuming – While the goal of utilising a chatbot is to speed up responses and improve customer contact, it appears to take longer due to the limited data provided and the time required for self-updating. While attending to a larger number of consumers at the same time, it appears befuddled.
- Zero decision-making — They are occasionally given the ability to make good decisions. Large corporations, such as Microsoft, are in the same boat.
- Why Chatbots are Difficult to Develop - Chatbots are developed using Natural Language Processing, which is a popular tool for customer assistance. Natural Language Processing is a subset of Machine Learning that may be used to communicate with users in textual form and respond to their questions. However, this necessitates extensive programming, which is difficult for businesses to accomplish. This is especially tough if organisations have to build chatbots from the ground up, which is why many web platforms make it simple for businesses to design and operate chatbots.

5. CONCLUSION

This research examines the tasks involved in NLU and ML, with the goal of incorporating them into chatbot systems to make them smarter. This report examines a number of research publications. This suggests that there will be a lot more research papers produced in this field in the future. It has been discovered that there are numerous algorithms for implementing all of the tasks required in NLU and ML. The proper algorithm to choose is determined by the chatbot's functionality as well as the domain in which the services will be given. In addition, while choosing an algorithm, the data format is critical.

By obtaining immediate assistance from the machine, the chatbot will aid in a pleasant customer experience. It will make customer service less stressful. In perilous situations, pre-programmed alarms will be useful. The question's reaction time is extremely short and precise. In the not-too-distant future, AI will take on a more prominent role and become a part of our daily lives. There is a constant need to seek out new ideas for improvement and to expand on already conducted research. A language model and a computational algorithm are combined in the chatbot architecture to follow information online contact between a human and a computer using common language. This computerized human-to-computer conversational plinths is hopefully working to deliver useful service in a variety of

industries to assist humans. The chatbot will be able to reply to queries outside of its dataset that are now occurring in the real world using APIs such as Government Services, Sports, Weather, and News. It is likely to meet large-scale information needs, wasting users' time and effort while increasing the process' competency and effectiveness. Chatterbots can be imagined as an assistant in any desired location in the near years, such as guiding for a specific task. Chatbots will help people overcome many of the problems they face at work. Furthermore, a chatbot can be viewed as a completely artificial intelligence-based robot capable of performing many tasks that would be impossible for a typical human to perform.

6. ACKNOWLEDGMENT

We would like to express our gratitude to all of the research paper authors that assisted us in significantly improving the work during the evaluation process. We also express our gratitude to all of the reference material's authors.

7. REFERENCES

[1] https://www.wordstream.com/blog/ws/2017/10/04/chatbots

[2] https://www.researchgate.net/publication/

[3] https://chatbotsmagazine.com/what-is-the-working-of-a-chatbot-e99e6996f51c

[4] https://en.wikipedia.org/wiki/Chatbot

[5] https://www.chatcompose.com/chatbot-learning.html

[6] https://www.ijtsrd.com/papers/ijtsrd31845.pdf

[7] Rincy Mariam Thomas, Supriya Punna, Mr. C Kishor Kumar Reddy, Dr. B V RamanaMurthy.(2019).Survey on Artificially Intelligent Chatbot. Journal of Applied Science and Computations, 1076-5131

[8] A Study of Today's A.I. through Chatbots and Rediscovery of Machine Intelligence.(ResearchGate)

[9] Intelligent Chatbot- https://www.ijert.org/intelligent-chatbot

[10] Diederich S., Brendel, A.B. Morana, S., and Kolbe L. (2021). "On the Desing of and Interaction with Conversational Agents: An Organizing and Assessing Review of Human-Computer Interaction Research".

[11] Feine J., Morana S., and Maedche A. (2020). "Designing Interactive Chatbot Development Systems".