CREATE A CHATBOT IN PYTHON

PHASE 5

ARTIFICIAL INTELLIGENCE CHATBOT USING PYTHON

Dr. C. K. GOMATHY[1], REDROUTHU VENKATA NARAYANA[2], THOTA VAMSI KRISHNA[3]

DR V GEETHA[4]

1, 4 -Assistant Professor, Department of Computer Science and Engineering SCSVMV, Kanchipuram

2, 3 Student- Department of Computer Science and Engineering, SCSVMV, Kanchipuram

ABSTRACT: A Chabot is artificial intelligence (AI) computer software that can simulate a conversation using textual or audio techniques. The basis of chat bots is artificial intelligence, which analyses a customer's data and blends the response 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. 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-Chabot connection.

KEYWORDS: Chabot, Chabot architecture, Artificial Intelligence, NLU (Natural language understanding), NLP (Natural language processing)

I.INTRODUCTION

Artificial Intelligence (AI) increasingly integrates our daily lives with the creation and analysis of intelligent software and hardware, called intelligent agents. Intelligent agents can do a variety of tasks ranging from labor work to sophisticated operations. A chatbot is a typical example of an AI system and one of the most elementary and widespread examples of intelligent Human-Computer Interaction (HCI). It is a computer program, which responds like a smart entity when conversed with through text or voice and understands one or more human languages by Natural Language Processing (NLP). Chatbots are also known as smart bots, interactive agents, digital assistants, or artificial conversation entities.

II. LITERATURE SURVEY

Alan Turing in 1950 proposed the Turing Test ("Can machines think?"), and it was at that time that the idea of a chatbot was popularized. The first known chatbot was Eliza, developed in 1966, whose purpose was to act as a psychotherapist returning the user utterances in a question form. It used simple pattern matching and a template-based response mechanism. Its conversational ability was not good, but it was enough to confuse people at a time when they were not used to interacting with computers and give them the impetus to start developing other chatbots. An improvement over ELIZA was a chatbot with a personality named PARRY developed in 1972. In 1995, the chatbot ALICE was developed which won the Loebnier Prize, an annual Turing Test, in years 2000, 2001, and 2004. It was the first computer to gain the rank of the "most human computer". ALICE relies on a simple pattern-matching algorithm with the underlying intelligence based on the Artificial Intelligence Markup Language (AIML), which makes it possible for developers to define the building blocks of the chatbot knowledge. Chatbots, like Smarter Child in 2001, were developed and became available through messenger applications. The next step was the creation of virtual personal assistants like Apple Siri, Microsoft Cortana, Amazon Alexa, Google Assistant and IBM Watson.

III. PROGRESS IN CHATBOT

A.A review on Chat Interface This unit is the front end of the system. It is responsible for collecting the user queries from the user which are the input to the system. It is also responsible for displaying the system generated results to the user. Therefore, it can be said that the chat interface is the face of the system through which the entire communication takes place. It is the mediator of conversation between the system and the user. The query that user fires on the chat interface is passed on to the chatting backend which acts as a message delivering system between the Chat interface and the Machine Learning Layer. This interface can be accessed either as a website or as a smart phone app. The type of interface depends on the requirements of the user that are to be satisfied by the system. If the system is accessed from a smartphone, the interface will be in the form of an app and if the system is accessed from a website, then the interface will be in the form of a website. For building apps on the smartphone, it will require to use android for android phones or Swift for iOS. In this case, only the interfacing platform will be programmed on android and the complete backend processing of the system will take place on a server on which the system will be deployed.

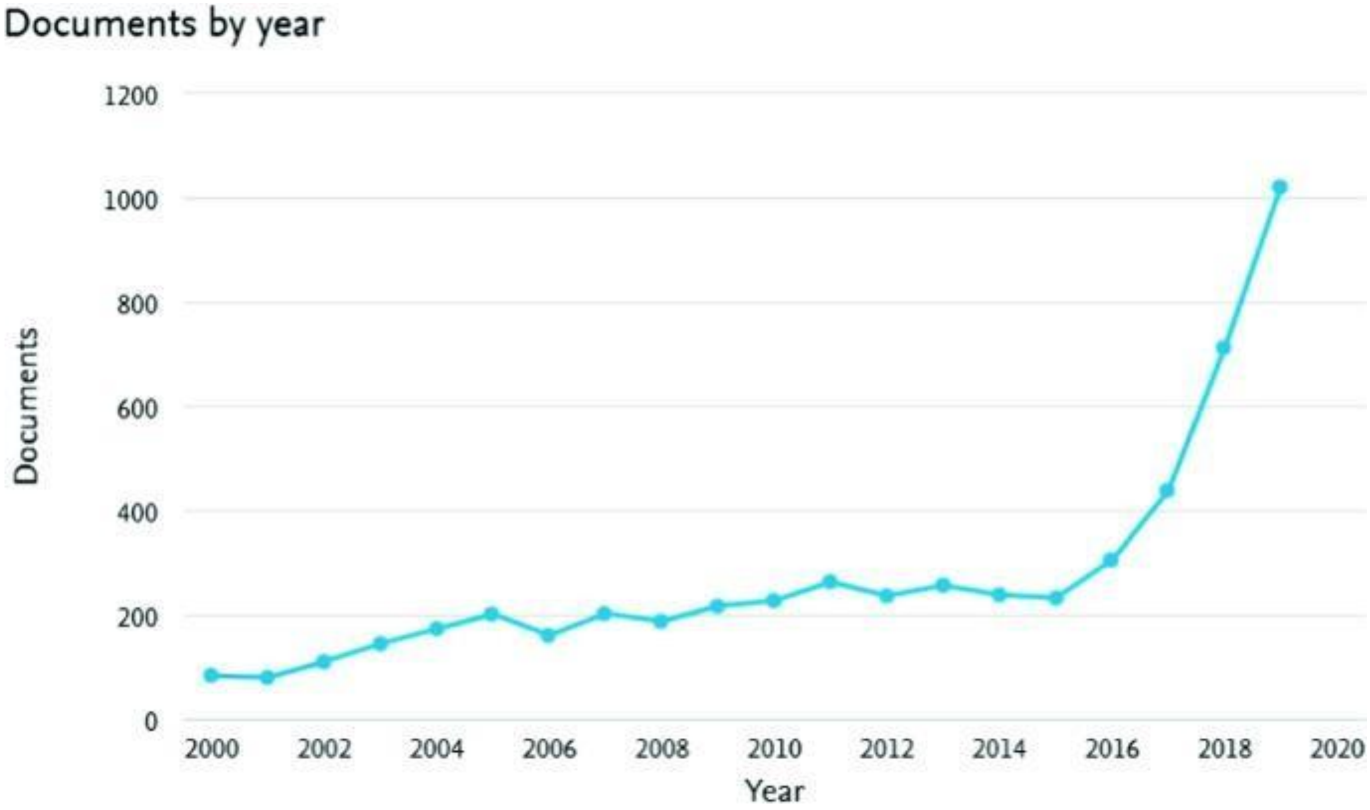


Fig 1: Development of Chabot

- Smart chat bots made up of NLU, NLG and ML engines have the following components:

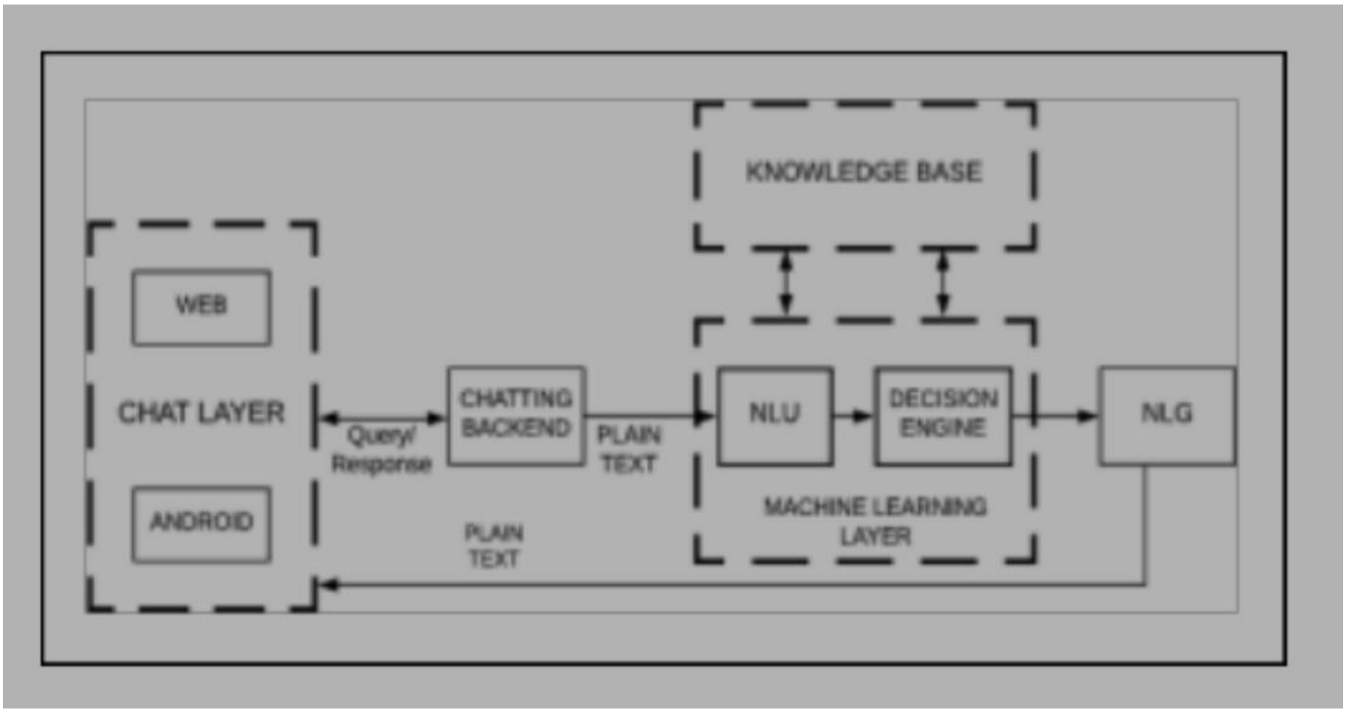


Fig 2: Architecture Of Chabot

B.**A review on NLU Engine** NLU i.e., Natural Language Understanding is a subpart of NLP (Natural Language Processing) which enables the system to understand the natural language or the conversation a language spoken by the users. The conversational language used by humans for day -day conversations is not as perfect the formal language. It does not focus much on the vocabulary and the grammar. Hence, it becomes difficult for a system to understand the intent of the sentence. The input received from the user is in unstructured text format which cannot be understood by the system directly. It understands input only in structured formats. The unstructured text received from the user is converted to structured format by extracting important words and patterns from the user text using the NLU techniques. Humans are capable of understanding any mispronunciations, homophones, swapped words, shortened form of words (like „its‟ for „it is‟), slang words or phrases and also words which are not used in formal vocabulary but exist in regular conversations.

C.**A review on Word Segmentation** also referred to as tokenization is the process of splitting text into smaller and meaningful units. These units could be paragraphs, sentences, clauses, phrases, words or letters. The smallest unit are the letters. Word segmentation is the splitting of sentences into individual words separated by blank spaces. The tokenized units of the sentences are called as tokens. The tokenizers split the sentences into words and punctuations marks as independent units. The most commonly used tokenizer is of space type, i.e., it splits the sentences into words at the blank spaces. It is also required that the tokenizer should consider abbreviations, acronyms, dates, numbers in decimal formats, etc., which cannot split at punctuations and blank spaces, as they will lose their meaning if done so.

D.**A review on POS Tagging** POS Tagging is the process of assigning grammatical annotations to individual words in the sentences. These annotations include the Parts-Of-Speech Tags. They denote the grammatical importance of the word in the sentence based on the dependency of that word with other words in that phrase, clause, sentence, paragraph, etc. The common POS tags are noun, verb, pronoun, etc. There are number of ways which can be used to perform the POS Tagging.

F.A review on Dependency Parsing A dependency parser is used to establish the relationship between words in a sentence based on the grammatical tags attached to it. It is the next step after parsing. A dependency tree or graph is created for every sentence. This tree is called as the parsing tree or the dependency tree. There are a number of ways by which the parsing can be implemented.

E.A review on Synonym and Pattern Recognition For information retrievals, no matter how big our data is, no sentence sent by the user can be perfectly same to any sentence in the database. But there can be sentences with the same intent. After understanding the intent of the user sentence, the database is checked for a sentence with the same intent. The matched sentences have difference of words which are used to express the same content.

G.A review on Decision or ML Engine Scripted or monotonous chatbots have predefined replies to be given. They provide replies to the user from a set of predefined replies categorized on the basis of the query given by the user. Inclusion of ML in chatbots enables it to compute the replies from scratch. It is used to make predictions to predict the responses for the user queries and also to update the system from its experiences. It keeps updating the databases as and when it encounters something new from the user. This engine uses supervised or unsupervised or both techniques to analyze what the user requires. It further uses a model to interpret the intent of the user and provides the appropriate results. The results may be in the form of predictions or any form of analysis which are based on the execution and analysis of mathematical models. Most of the machine learning models are based on statistical and probabilistic evaluations of the instances occurring and the calculations makes a prediction for the test instance.

H.A review on NLG Engine NLG performs the reverse task of NLU. It is the process of converting the system produced results into natural language representations which can be easily understood by the user. In other words, NLG is the process of generating text/speech from system generated patterns. The results produced by the system are in the structured format so that they can be easily understood and processed by the system. NLG represents the System knowledge base in a natural or conversational language representation which can be easily understood by the user. There can be a number of ways in which in which a same sentence can be said. The sentences can have two voices i.e., active or passive voice. Also, there can be similarity between two sentences, but they might involve the usage of synonyms. Hence, while providing a response to the user, the NLG unit needs to calculate all the possibilities to interpret the same sentence, and then select the most appropriate one.

IV. PROPOSED SYSTEM

- **Context identification:** The input text is pre-processed to standardize it according to the system's requirements. The proper context is detected based on the keywords used in the text.
- **AIML Response system:** the user is attempting to have a normal conversation with the bot, and the input is mapped to an appropriate pattern in AIML files. The user receives the response if it is available.
- **Query Analysis and Response System:** upon receiving the personal queries like hey, hello, what are you doing?. The input text is analyzed to extract keywords. The user's request for information is understood based on the keywords, and the information is retrieved from the database.

1. When a user wants a information that he needs that will be provided through this module. If the user's input matches a pattern in the AIML files, the user will receive the appropriate response. If the AIML files do not contain an item for that query pattern, keywords are retrieved from the input.
2. For suppose, if the sentence is retrieved with confidence > 0.5 , we return the answer of that question as the response. And if the bot was not understand what the user is given to it will generates it is understand, give some information, like that.
3. Once the user is satisfied with the response which is generated by the bot and does not wish to chat further, he/she has the option to say goodbye and terminate the activity.

Advantages

- **Real time** responses to all recurring questions
- **Available** 24\7
- Conversational intelligence, personalization
- **Web, mobile, messaging** applications implementation
- Eliminate tedious time-consuming tasks

V. SYSTEM ARCHITECTURE

Typical chatbot architecture should consist of the following:

- Chat window/ session/ or frontend application interface.
- The deep learning model for Natural Language Processing [NLP].
- Corpus or training data for training the NLP model Application Database for processing actions to be performed by the chatbot.

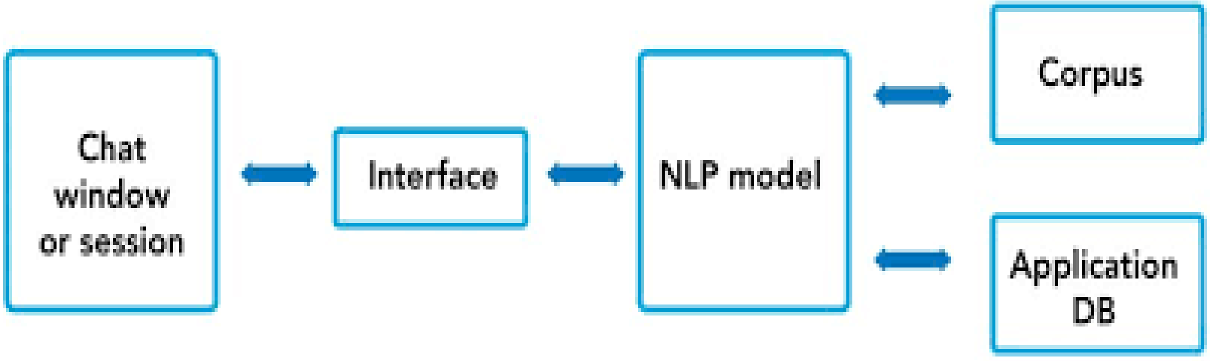


Fig 3: System Architecture

A) Corpus or Training Data: Corpus means the data that could be used to train the NLP model to understand the human language as text or speech and reply using the same medium. Corpus is usually huge data with a lot of human interactions.

B) Chatbot window: We have designed a function which enables the user to interact with bot using text. The function keeps the chat window alive unless it is asked to break or quit. The name of our text bot is Jason.
C) Evaluate or test the chatbot: There could be multiple paths using which we can interact and evaluate the built text bot.

D) Data pre-processing
Text case [upper or lower] handling: Convert all the data coming as an input [corpus or user inputs] to either upper or lower case. This will avoid misrepresentation and misinterpretation of words if spelt under lower or upper cases.

E) Tokenization: Convert a sentence [i.e., a collection of words] into single words.

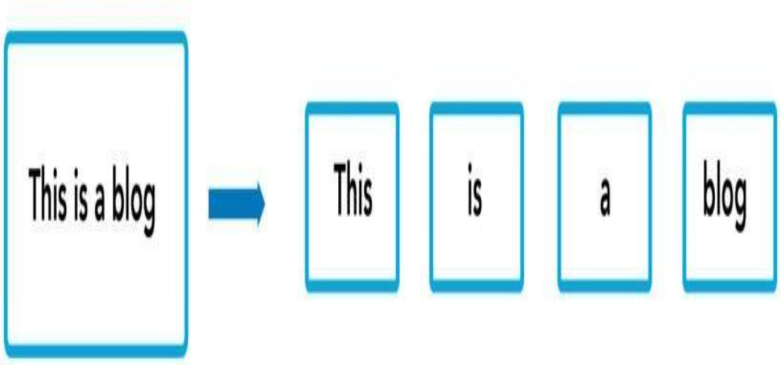
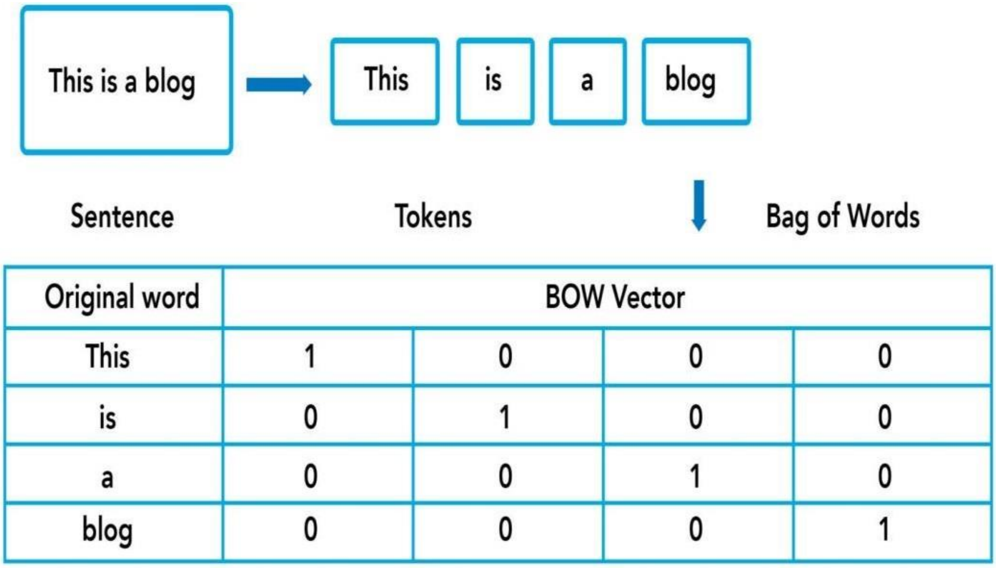


Fig 4: Tokenization process

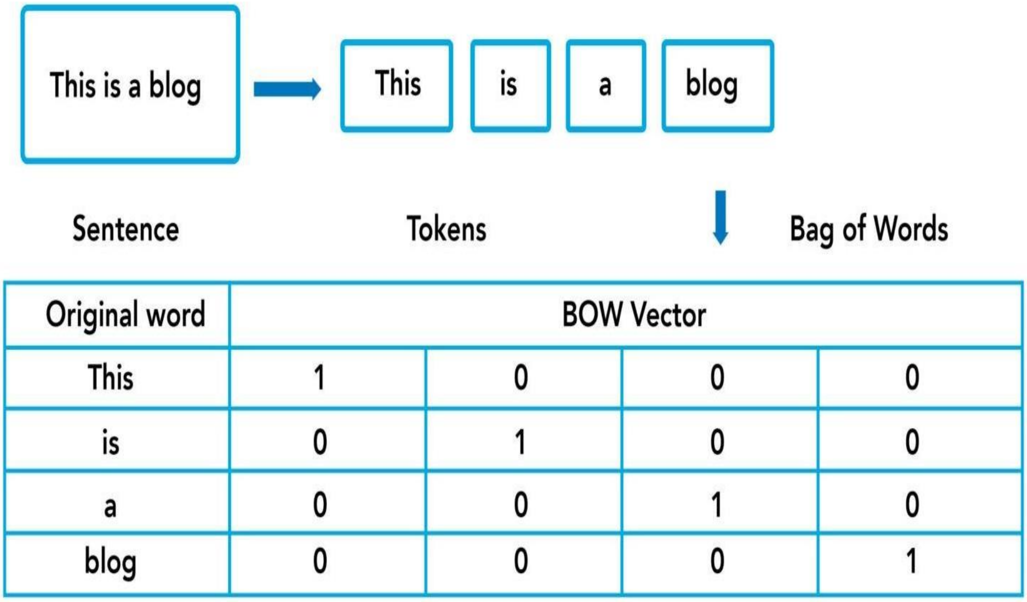
F) Stemming: It is a process to find similarities between words with the same root words. This will help us to reduce the bag of words by associating similar words with their corresponding root words.



Original word	BOW Vector			
This	1	0	0	0
is	0	1	0	0
a	0	0	1	0
blog	0	0	0	1

Fig 5: Stemming process

G) Generate BOW [Bag of Words]: Process of converting words into numbers by generating vector embeddings from the tokens generated above. This is given as an input to the neural network model for understanding the written text.



Original word	BOW Vector			
This	1	0	0	0
is	0	1	0	0
a	0	0	1	0
blog	0	0	0	1

Fig 5: Bag of words

VI. MODULES

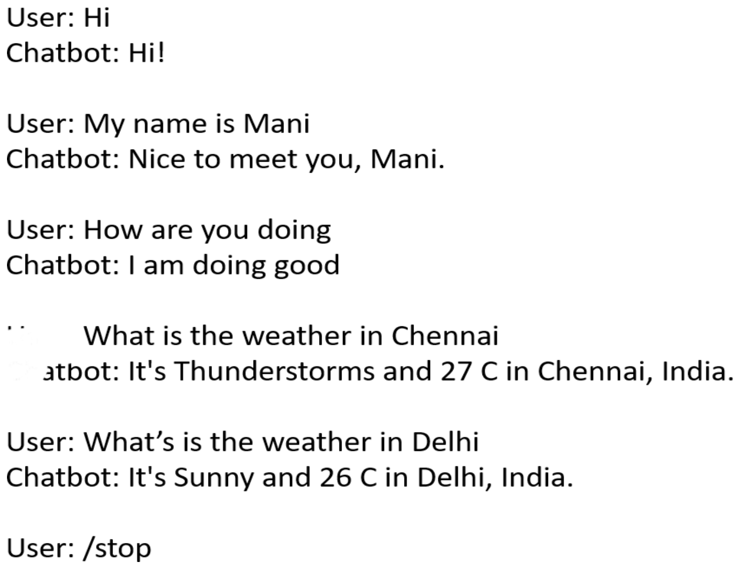
It has three python code modules.

- ❑ **Intents.json:**intents classification or recognization it is a type of getting a spoken or written text and then classifying it based on what the user wants to achieve.
- ❑ **Trainer.py:** Defines the Chatbot overall file structure and contains the intent, actions, slots, stories, domain, config and endpoint details. The code will train an NLU and dialogue model to retrieve weather from the Yahoo weather API. Model folder contains the trained models. It will also start the server with actions and also runs the chatbot on the command line. Execute only this code as it will trigger the actions and run.py.
- ❑ **Run.py:** triggered by trainer.py. contains the modules to run the chatbot module in the command line.

IMPLEMENTATION

- ➢ The implementations of a chatbot involve a variety of techniques. Understanding what the chatbot will offer and what category falls into helps developers pick the algorithms or platforms and tools to build it. At the same time, it also helps the end-users understand what to expect.
- ➢ The requirements for designing a chatbot include accurate knowledge representation, an answer generation strategy, and a set of predefined neutral answers to reply when user utterance is not understood. The first step in designing any system is to divide it into constituent parts according to a standard so that a modular development approach can be followed.
- ➢ The process starts with a user's request, for example, "What is the meaning of environment?", to the chatbot using a messenger app like Facebook, Slack, WhatsApp, WeChat or Skype, or an app using text or speech input like Amazon Echo.
- ➢ After the chatbot receives the user request, the Language Understanding Component parses it to infer the user's intention and the associated information (intent: "translate," entities: [word: "environment"]).
- ➢ Once a chatbot reaches the best interpretation it can, it must determine how to proceed. It can act upon the new information directly, remember whatever it has understood and wait to see what happens next, require more context information or ask for clarification.
- ➢ When the request is understood, action execution and information retrieval take place. The chatbot performs the requested actions or retrieves the data of interest from its data sources, which may be a database, known as the Knowledge Base of the chatbot, or external resources that are accessed through an API call.
- ➢ Upon retrieval, the Response Generation Component uses Natural Language Generation (NLG) to prepare a natural language human-like response to the user based on the intent and context information returned from the user message analysis component. The appropriate responses are produced by one of the three models: rule-based, retrieval based, and generative model.
- ➢ A Dialogue Management Component keeps and updates the context of a conversation which is the current intent, identified entities, or missing entities required to fulfill user requests. Moreover, it requests missing information, processes clarifications by users, and asks follow-up questions. For

- example, the chatbot may respond: “Would you like to tell me as well an example sentence with the word environment?”.



```
User: Hi
Chatbot: Hi!

User: My name is Mani
Chatbot: Nice to meet you, Mani.

User: How are you doing
Chatbot: I am doing good

        What is the weather in Chennai
  atbot: It's Thunderstorms and 27 C in Chennai, India.

User: What’s is the weather in Delhi
Chatbot: It's Sunny and 26 C in Delhi, India.

User: /stop
```

Fig 6: Chatting process

VII. RESULT

The purpose of a chatbot system is to simulate a human conversation. Its architecture integrates a language model and computational algorithm to emulate information online communication between a human and a computer using natural language.

VIII. CONCLUSION

chatbots or smart assistants with artificial intelligence, in my opinion, are revolutionizing the world. In comparison to larger chatbots developing a simple chatbot is not a difficult effort, and developers need understand and address concerns such as reliability, scalability, and adaptability, as well as high level of intent on human language. Chatbots are more effective than people in reaching out to a big audience via messaging apps. They have the potential to become a useful information gathering tool in the near future.