A Project/Dissertation Report

on

BANK CHATBOT USING PYTHON

Submitted in partial fulfillment of the

requirement for the award ofthe degree of

COMPUTER SCIENCE AND ENGINEERING SCHOOL OF COMPUTING SCIENCE AND ENGINEERING

(Established under Galgotias University Uttar Pradesh Act No. 14 of 2011)

Under The Supervision of
Mr A.Arul Prakash
Assitant Professor

Submitted By

Lakshay Kumar(19021011372)

Vishwadeep Rana(19021011426)

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
GALGOTIAS UNIVERSITY, GREATER NOIDA
INDIA
DECEMBER, 2021

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING
GALGOTIAS UNIVERSITY, GREATER NOIDA

CANDIDATE'S DECLARATION

We hereby certify that the work which is being presented in the thesis/project/dissertation, entitled **"BANK CHATBOT USING PYTHON"** in partial fulfillment of the requirements for the award of the Bachelor of Technology submitted in the School of Computing Science and Engineering of Galgotias University, Greater Noida, is an original work carried out during the period of December, 2021, under the supervision of Mr A.Arul Prakash Assitant Professor, Department of Computer Science and Engineering/Computer Application and Information and Science, of School of Computing Science and Engineering, Galgotias University, Greater Noida.

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Mr A.Arul Prakash

Assitant Professor

____

<u>CERTIFICATE</u>

The Final Thesis/Project/ Dissertation Viva-Voce examination of Lakshay Kumar 19SCSE1010178 and Vishwadeep Rana 19SCSE1010237 has been held on _________________ and their work is recommended for the award of Bachelor of Technology.

Signature of Examiner(s) **Signature of Supervisor(s)**

Signature of Project Coordinator **Signature of Dean**

Date: December, 2021

Place: Greater Noida

Abstract

Chat bots are the system or you can say a software that act as a normal person to solve your problems or deal with the situations that you are facing off. So, we are making a project on the bank chat bot. Which will help the users of the bank for doing any tramp. It helps like a real person such as we go to a bank for any help and we contact a person for this. But this bank chat bot will help us instead of a person to solve our worry .So, by remaining in our homes only we can get benefitted. It is a very handy software in nowadays online systems. Due to the COVID-19 people cannot get out so, people can use this initiative to solve their problems that they are facing with their banks. Our chat bot system can do every type of work that a bank can do such withdraw , deposit, knowing about the bank and each item . So it is a most beneficiary thing in our lifestyle.

List of Tables

List of Figures

Acronyms

B.Tech.	Bachelor of Technology
M.Tech.	Master of Technology
BCA	Bachelor of Computer Applications
MCA	Master of Computer Applications
B.Sc. (CS)	Bachelor of Science in Computer Science
M.Sc. (CS)	Master of Science in Computer Science
SCSE	School of Computing Science and Engineering

TABLE OF CONTENTS

LIST OF SYMBOLS, ABBREVIATIONS AND NOMENCLATURE

ACH	AUTOMATED CLEARING HOUSE
ISP	INTERNET SERVICE PROVIDER
OCC	OPEN CASH CREDIT
ATM	AUTOMATIC TELLER MACHINE
JDBC	JAVA DATABASE CONNECTIVITY
JSP	JAVA SERVER PAGE
IDE	INTEGRATED DEVELOPMENT ENVIRONMENT
DFD	DATA FLOW DIAGRAM
UML	UNIFIED MODELING LANGUAGE
SQL	STRUCTURE QUERY LANGUAGE
DML	DATA MANIPULATION LANGUAGE
GUI	GRAPHIC USER INTERFACE
DBS	DATABASE SYSTEMS
J2EE	JAVA 2 PLATFORM ENTERPRISE EDITION
J2SDKD	JAVA 2 SDKD
JVM	JAVA VIRTUAL MACHINE
CGI	COMMON GATEWAY INTERFACE

CHAPTER-1 Introduction

1.1 Chatbot and Machine learning

Machine learning chatbots works using artificial intelligence. User need not to be more specific while talking with a bot because it can understand the natural language, not only commands. This kind of bots get continuously better or smarter as it learns from past conversations it had with people.
Here is a simple example which illustrate how they work. The following is a conversation between a human and a chatbot: Human: “What is a bank.” Bot: “what is a bank While a bank is a profit-driven entity, a credit union is a nonprofit organization traditionally run by volunteers. A bank is a financial institution licensed to receive deposits and make loans.”

1.2 Artificial Intelligence

AI was coined by John McCarthy, an American computer scientist, in 1956 at The Dartmouth Conference where the discipline was born. Today, it is an umbrella term that encompasses everything from robotic process automation to actual robotics. It has gained prominence recently due, in part, to big data, or the increase in speed, size and variety of data businesses are now collecting. AI can perform tasks such as identifying patterns in the data more efficiently than humans, enabling businesses to gain more insight out of their data. As machines become increasingly capable, tasks considered to require "intelligence" are often removed from the definition of AI, a phenomenon known as the AI effect. A quip in Tesler's Theorem says "AI is whatever hasn't been done yet." For instance, optical character recognition is frequently excluded from things considered to be AI, having become a routine technology. Modern machine capabilities generally classified as AI include successfully understanding human speech, competing at the highest level in strategic game systems (such as chess and Go), and also imperfect-information games like poker, self-driving cars, intelligent routing in content delivery networks, and military simulations.
Thought-capable artificial beings appeared as storytelling devices in antiquity, and have been common in fiction, as in Mary Shelley's Frankenstein or Karel Čapek's R.U.R. These
characters and their fates raised many of the same issues now discussed in the ethics of artificial intelligence.

The study of mechanical or "formal" reasoning began with philosophers and mathematicians in antiquity. The study of mathematical logic led directly to Alan Turing's theory of computation, which suggested that a machine, by shuffling symbols as simple as "0" and "1", could simulate any conceivable act of mathematical deduction. This insight, that digital computers can simulate any process of formal reasoning, is known as the Church–Turing thesis. Along with concurrent discoveries in neurobiology, information theory and cybernetics, this led researchers to consider the possibility of building an electronic brain. Turing proposed changing the question from whether a machine was intelligent, to "whether or not it is possible for machinery to show intelligent behaviour". The first work that is now generally recognized as AI was McCullouch and Pitts' 1943 formal design for Turing-complete "artificial neurons".

The field of AI research was born at a workshop at Dartmouth College in 1956, where the term "Artificial Intelligence" was coined by John McCarthy to distinguish the field from cybernetics and escape the influence of the cyberneticist Norbert Wiener. Attendees Allen Newell (CMU), Herbert Simon (CMU), John McCarthy (MIT), Marvin Minsky (MIT) and Arthur Samuel (IBM) became the founders and leaders of AI research. They and their students produced programs that the press described as "astonishing": computers were learning checkers strategies (c. 1954) (and by 1959 were reportedly playing better than the average human), solving word problems in algebra, proving logical theorems (Logic Theorist, first run c. 1956) and speaking English. By the middle of the 1960s, research in the U.S. was heavily funded by the Department of Defense and laboratories had been established around the world. AI's founders were optimistic about the future: Herbert Simon predicted, "machines will be capable, within twenty years, of doing any work a man can do". Marvin Minsky agreed, writing, "within a generation ... the problem of creating 'artificial intelligence' will substantially be solved". They failed to recognize the difficulty of some of the remaining tasks. Progress slowed and in 1974, in response to the criticism of Sir James Lighthill and ongoing pressure from the US Congress to fund more productive projects, both the U.S. and British governments cut off exploratory research in AI. The next few years would later be called an "AI winter", a period when obtaining funding for AI projects was difficult. In the early 1980s, AI research was revived by the commercial success of expert systems, a form of AI program that simulated the knowledge and analytical skills of human experts. By 1985, the market for AI had reached over a billion dollars. At the same time, Japan's fifth generation computer project inspired the U.S and British governments to restore funding for academic research. However, beginning with the collapse of the Lisp Machine market in 1987, AI once again fell into disrepute, and a second, longer-lasting hiatus began.

The development of metal–oxide–semiconductor (MOS) very-large-scale integration (VLSI), in the form of complementary MOS (CMOS) transistor technology, enabled the development of practical artificial neural network (ANN) technology in the 1980s. A landmark publication in the field was the 1989 book Analog VLSI Implementation of Neural Systems by Carver A. Mead and Mohammed Ismail.

In the late 1990s and early 21st century, AI began to be used for logistics, data mining, medical diagnosis and other areas. The success was due to increasing computational power (see Moore's law and transistor count), greater emphasis on solving specific problems, new ties between AI and other fields (such as statistics, economics and mathematics), and a commitment by researchers to mathematical methods and scientific standards. Deep Blue became the first computer chess-playing system to beat a reigning world chess champion, Garry Kasparov, on 11 May 1997.

In 2011, in a Jeopardy! quiz show exhibition match, IBM's question answering system, Watson, defeated the two greatest Jeopardy! champions, Brad Rutter and Ken Jennings, by a significant margin. Faster computers, algorithmic improvements, and access to large amounts of data enabled advances in machine learning and perception; data-hungry deep learning methods started to dominate accuracy benchmarks around 2012. The Kinect, which provides a 3D body– motion interface for the Xbox 360 and the Xbox One, uses algorithms that emerged from lengthy AI research as do intelligent personal assistants in smartphones. In March 2016, AlphaGo won 4 out of 5 games of Go in a match with Go champion Lee Sedol, becoming the first computer Go-playing system to beat a professional Go player without handicaps. In the 2017 Future of Go Summit, AlphaGo won a three-game match with Ke Jie, who at the time continuously held the world No. 1 ranking for two years. Deep Blue's Murray Campbell called AlphaGo's victory "the end of an era... board games are more or less done and it's time to move on." This marked the completion of a significant milestone in the development of Artificial Intelligence as Go is a relatively complex game, more so than Chess.

AlphaGo was later improved, generalized to other games like chess, with AlphaZero; and MuZero to play many different video games, that were previously handled separately, in addition to board games. Other programs handle imperfect-information games; such as for poker at a superhuman level, Pluribus (poker bot) and Cepheus (poker bot). See: General game playing.

According to Bloomberg's Jack Clark, 2015 was a landmark year for artificial intelligence, with the number of software projects that use AI within Google increased from a "sporadic usage" in 2012 to more than 2,700 projects. Clark also presents factual data indicating the improvements of AI since 2012 supported by lower error rates in image processing tasks.

He attributes this to an increase in affordable neural networks, due to a rise in cloud computing infrastructure and to an increase in research tools and datasets. Other cited examples include Microsoft's development of a Skype system that can automatically translate from one language to another and Facebook's system that can describe images to blind people. In a 2017 survey, one in five companies reported they had "incorporated AI in some offerings or processes". Around 2016, China greatly accelerated its government funding; given its large supply of data and its rapidly increasing research output, some observers believe it may be on track to becoming an "AI superpower".

By 2020, Natural Language Processing systems such as the enormous GPT-3 (then by far the largest artificial neural network) were matching human performance on pre-existing benchmarks, albeit without the system attaining commonsense understanding of the contents of the benchmarks.

DeepMind's AlphaFold 2 (2020) demonstrated the ability to determine, in hours rather than months, the 3D structure of a protein. Facial recognition advanced to where, under some circumstances, some systems claim to have a 99% accuracy rate.

Basics Computer science defines AI research as the study of "intelligent agents": any device that perceives its environment and takes actions that maximize its chance of successfully achieving its goals. A more elaborate definition characterizes AI as "a system's ability to correctly interpret external data, to learn from such data, and to use those learnings to achieve specific goals and tasks through flexible adaptation."

A typical AI analyzes its environment and takes actions that maximize its chance of success. An AI's intended utility function (or goal) can be simple ("1 if the AI wins a game of Go, 0 otherwise") or complex ("Perform actions mathematically similar to ones that succeeded in the past"). Goals can be explicitly defined or induced. If the AI is programmed for "reinforcement learning", goals can be implicitly induced by rewarding some types of behavior or punishing others.

Alternatively, an evolutionary system can induce goals by using a "fitness function" to mutate and preferentially replicate high-scoring AI systems, similar to how animals evolved to innately desire certain goals such as finding food. Some AI systems, such as nearest- neighbor, instead of reason by analogy, these systems are not generally given goals, except to the degree that goals are implicit in their training data. Such systems can still be benchmarked if the non-goal system is framed as a system whose "goal" is to accomplish its narrow classification task.

AI often revolves around the use of algorithms. An algorithm is a set of unambiguous instructions that a mechanical computer can execute. A complex algorithm is often built on top of other, simpler, algorithms.

A simple example of an algorithm is the following (optimal for first player) recipe for play at tic-tac-toe: If someone has a "threat" (that is, two in a row), take the remaining square. Otherwise, if a move "forks" to create two threats at once, play that move. Otherwise, take the center square if it is free. Otherwise, if your opponent has played in a corner, take the opposite corner. Otherwise, take an empty corner if one exists. Otherwise, take any empty square.

Many AI algorithms are capable of learning from data; they can enhance themselves by learning new heuristics (strategies, or "rules of thumb", that have worked well in the past), or can themselves write other algorithms. Some of the "learners" described below, including Bayesian networks, decision trees, and nearest-neighbor, could theoretically, (given infinite data, time, and memory) learn to approximate any function, including which combination of mathematical functions would best describe the world.[citation needed] These learners could therefore derive all possible knowledge, by considering every possible hypothesis and matching them against the data. In practice, it is seldom possible to consider every possibility, because of the phenomenon of "combinatorial explosion", where the time needed to solve a problem grows exponentially. Much of AI research involves figuring out how to identify and avoid considering a broad range of possibilities unlikely to be beneficial. For example, when viewing a map and looking for the shortest driving route from Denver to New York in the East, one can in most cases skip looking at any path through San Francisco or other areas far to the West; thus, an AI wielding a pathfinding algorithm like A* can avoid the combinatorial explosion that would ensue if every possible route had to be ponderously considered.

The earliest (and easiest to understand) approach to AI was symbolism (such as formal logic): "If an otherwise healthy adult has a fever, then they may have influenza". A second, more general, approach is Bayesian inference: "If the current patient has a fever, adjust the probability they have influenza in such-and-such way".

The third major approach, extremely popular in routine business AI applications, are analogizers such as SVM and nearest-neighbor: "After examining the records of known past patients whose temperature, symptoms, age, and other factors mostly match the current patient, X% of those patients turned out to have influenza". A fourth approach is harder to intuitively understand, but is inspired by how the brain's machinery works: the artificial neural network approach uses artificial "neurons" that can learn by comparing itself to the desired output and altering the strengths of the connections between its internal neurons to "reinforce" connections that seemed to be useful. These four main approaches can overlap with each other and with evolutionary systems; for example, neural nets can learn to make inferences, to generalize, and to make analogies. Some systems implicitly or explicitly use multiple of these

approaches, alongside many other AI and non-AI algorithms; the best approach is often different depending on the problem.
Learning algorithms work on the basis that strategies, algorithms, and inferences that worked well in the past are likely to continue working well in the future. These inferences can be obvious, such as "since the sun rose every morning for the last 10,000 days, it will probably rise tomorrow morning as well". They can be nuanced, such as "X% of families have geographically separate species with color variants, so there is a Y% chance that undiscovered black swans exist". Learners also work on the basis of "Occam's razor": The simplest theory that explains the data is the likeliest. Therefore, according to Occam's razor principle, a learner must be designed such that it prefers simpler theories to complex theories, except in cases where the complex theory is proven substantially better.

Types of artificial intelligence

AI can be categorized in any number of ways, but here are two examples.
The first classifies AI systems as either weak AI or strong AI. Weak AI, also known as narrow AI, is an AI system that is designed and trained for a particular task. Virtual personal assistants, such as Apple's Siri, are a form of weak AI.
Strong AI, also known as artificial general intelligence, is an AI system with generalized human cognitive abilities so that when presented with an unfamiliar task, it has enough intelligence to find a solution. The Turing Test, developed by mathematician Alan Turing in 1950, is a method used to determine if a computer can actually think like a human, although the method is controversial.

The second example is from Arend Hintze, an assistant professor of integrative biology and computer science and engineering at Michigan State University. He categorizes AI into four types, from the kind of AI systems that exist today to sentient systems, which do not yet exist. His categories are as follow:-

Type 1: Reactive machines.
An example is Deep Blue, the IBM chess program that beat Garry Kasparov in the 1990s. Deep Blue can identify pieces on the chess board and make predictions, but it has no memory and cannot use past experiences to inform future ones. It analyzes possible moves -- its own and its opponent -- and chooses the most strategic move. Deep Blue and Google's AlphaGO were designed for narrow purposes and cannot easily be applied to another situation.

Type 2: Limited memory.

These AI systems can use past experiences to inform future decisions. Some of the decision- making functions in autonomous vehicles have been designed this way. Observations used to inform actions happening in the not-so-distant future, such as a car that has changed lanes. These observations are not stored permanently.

Type 3: Theory of mind.

This is a psychology term. It refers to the understanding that others have their own beliefs, desires and intentions that impact the decisions they make. This kind of AI does not yet exist.

Type 4: Self-awareness. In this category,

AI systems have a sense of self, have consciousness. Machines with self-awareness understand their current state and can use the information to infer what others are feeling. This type of AI does not yet exist.

Examples of AI technology

Automation is the process of making a system or process function automatically. Robotic process automation, for example, can be programmed to perform high-volume, repeatable tasks normally performed by humans. RPA is different from IT automation in that it can adapt to changing circumstances.

Machine learning is the science of getting a computer to act without programming. Deep learning is a subset of machine learning that, in very simple terms, can be thought of as the automation of predictive analytics. There are three types of machine learning algorithms: supervised learning, in which data sets are labeled so that patterns can be detected and used to label new data sets; unsupervised learning, in which data sets aren't labeled and are sorted according to similarities or differences; and reinforcement learning, in which data sets aren't labeled but, after performing an action or several actions, the AI system is given feedback.

Machine vision is the science of making computers see. Machine vision captures and analyzes visual information using a camera, analog-to-digital conversion and digital signal processing. It is often compared to human eyesight, but machine vision isn't bound by biology and can be programmed to see through walls, for example. It is used in a range of applications from signature identification to medical

image analysis. Computer vision, which is focused on machine-based image processing, is often conflated with machine vision.
Natural language processing (NLP) is the processing of human -- and not computer -- language by a computer program. One of the older and best known examples of NLP is spam detection, which looks at the subject line and the text of an email and decides if it's junk. Current approaches to NLP are based on machine learning. NLP tasks include text translation, sentiment analysis and speech recognition.
Pattern recognition is a branch of machine learning that focuses on identifying patterns in data. The term, today, is dated.
Robotics is a field of engineering focused on the design and manufacturing of robots. Robots are often used to perform tasks that are difficult for humans to perform or perform consistently. They are used in assembly lines for car production or by NASA to move large objects in space. More recently, researchers are using machine learning to build robots that can interact in social settings.

1.3 AI applications

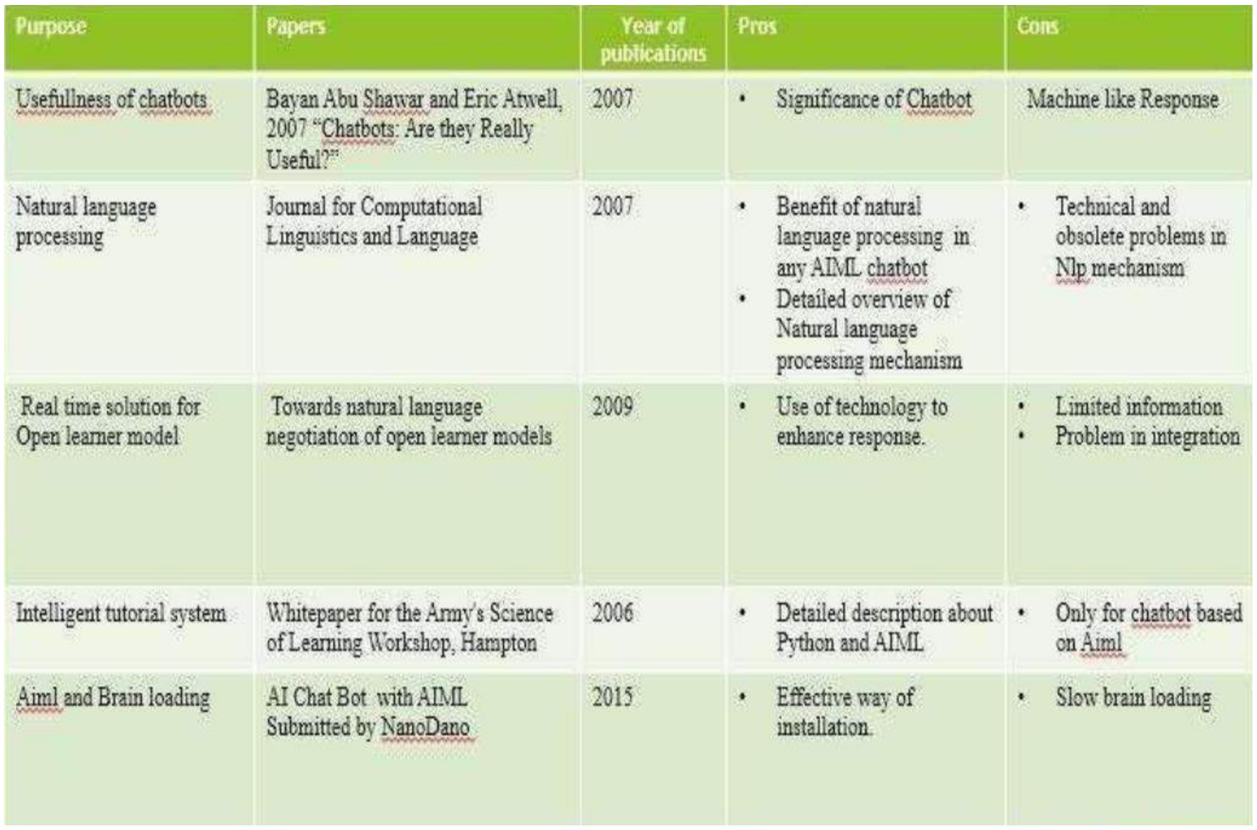
AI in healthcare. The biggest bets are on improving patient outcomes and reducing costs. Companies are applying machine learning to make better and faster diagnoses than humans. One of the best known healthcare technologies is IBM Watson. It understands natural language and is capable of responding to questions asked of it. The system mines patient data and other available data sources to form a hypothesis, which it then presents with a confidence scoring schema.

Other AI applications include chatbots, a computer program used online to answer questions and assist customers, to help schedule follow-up appointments or aiding patients through the billing process, and virtual health assistants that provide basic medical feedback.
AI in business. Robotic process automation is being applied to highly repetitive tasks normally performed by humans. Machine learning algorithms are being integrated into analytics and CRM platforms to uncover information on how to better serve customers. Chatbots have been incorporated into websites to provide immediate service to customers. Automation of job positions has also become a talking point among academics and IT consultancies such as Gartner and Forrester.
AI in education. AI can automate grading, giving educators more time. AI can assess students and adapt to their needs, helping them work at their own pace. AI tutors can provide additional support to students, ensuring they stay on track. AI could change where and how students learn, perhaps even replacing some teachers.

AI in finance. AI applied to personal finance applications, such as Mint or Turbo Tax, is upending financial institutions. Applications such as these could collect personal data and provide financial advice. Other programs, IBM Watson being one, have been applied to the process of buying a home. Today, software performs much of the trading on Wall Street. AI in law. The discovery process, sifting through of documents, in law is often overwhelming for humans. Automating this process is a better use of time and a more efficient process. Startups are also building question-and-answer computer assistants that can sift programmed-to-answer questions by examining the taxonomy and ontology associated with a database.

AI in manufacturing. This is an area that has been at the forefront of incorporating robots into the workflow. Industrial robots used to perform single tasks and were separated from human workers, but as the technology advanced that changed.

CHAPTER-2 Literature Reviews



Purpose	Papers	Year of publications	Pros	Cons
Usefullness of chatbots	Bayan Abu Shawar and Eric Atwell, 2007 "Chatbots: Are they Really Useful?"	2007	• Significance of Chatbot	Machine like Response
Natural language processing	Journal for Computational Linguistics and Language	2007	• Benefit of natural language processing in any AIML chatbot • Detailed overview of Natural language processing mechanism	• Technical and obsolete problems in Nlp mechanism
Real time solution for Open learner model	Towards natural language negotiation of open learner models	2009	• Use of technology to enhance response.	• Limited information • Problem in integration
Intelligent tutorial system	Whitepaper for the Army's Science of Learning Workshop, Hampton	2006	• Detailed description about Python and AIML	• Only for chatbot based on Aiml
Aiml and Brain loading	AI Chat Bot with AIML Submitted by NanoDano	2015	• Effective way of installation.	• Slow brain loading

<u>DFD DIAGRAM OF CHAT BOT:-</u>

Fig:1

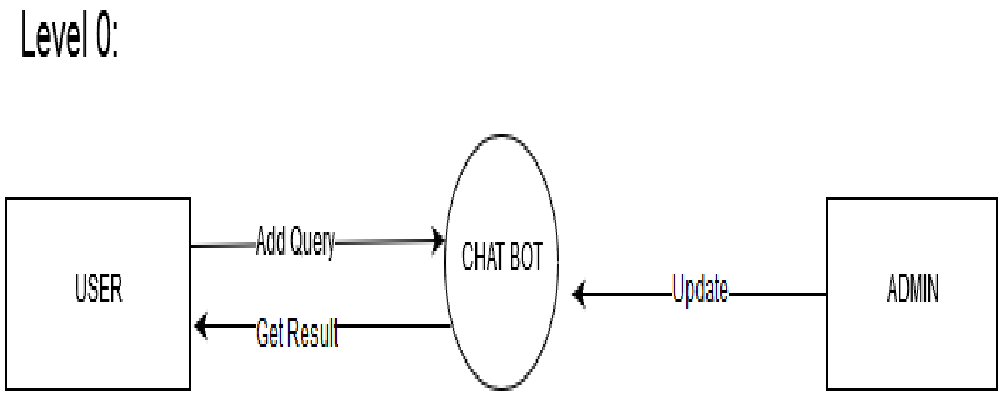
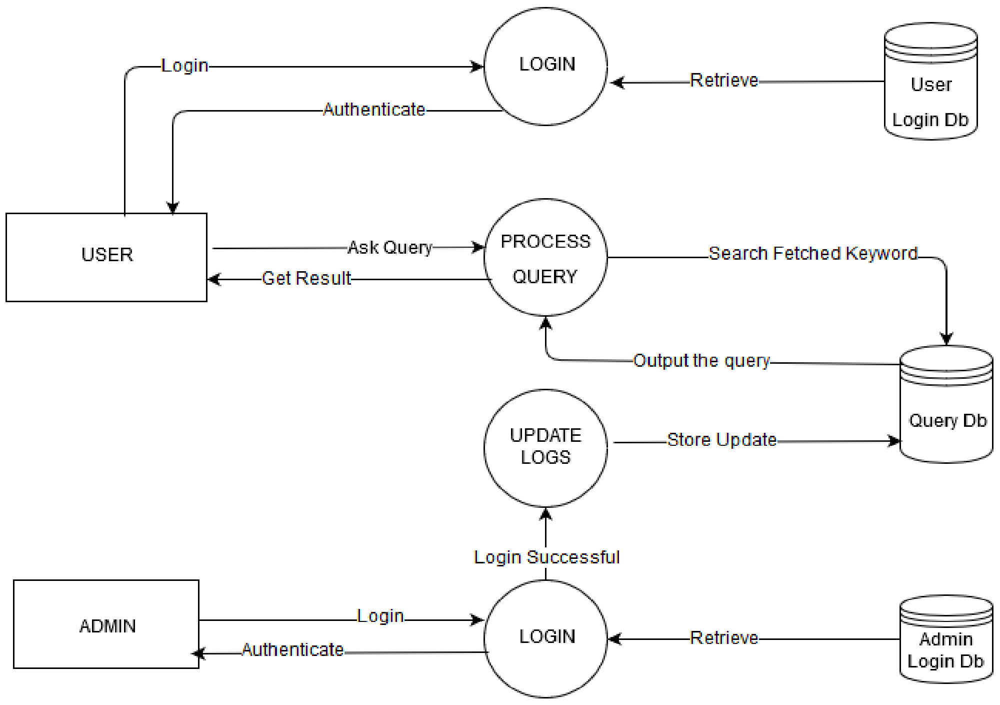


Fig 2:

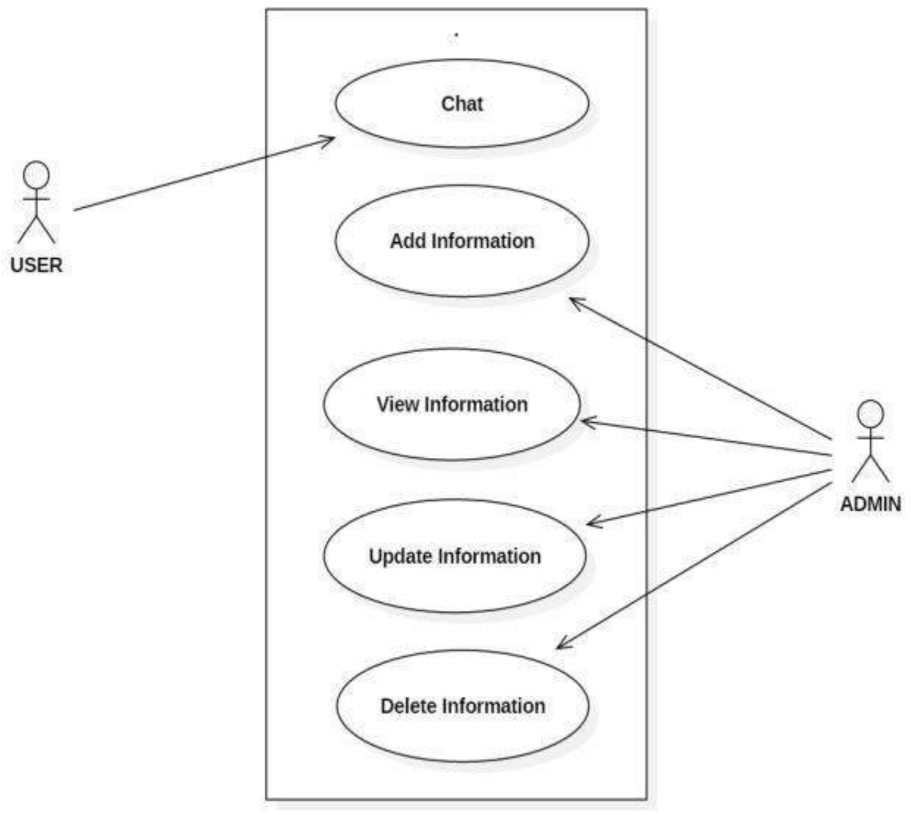
Level 1:



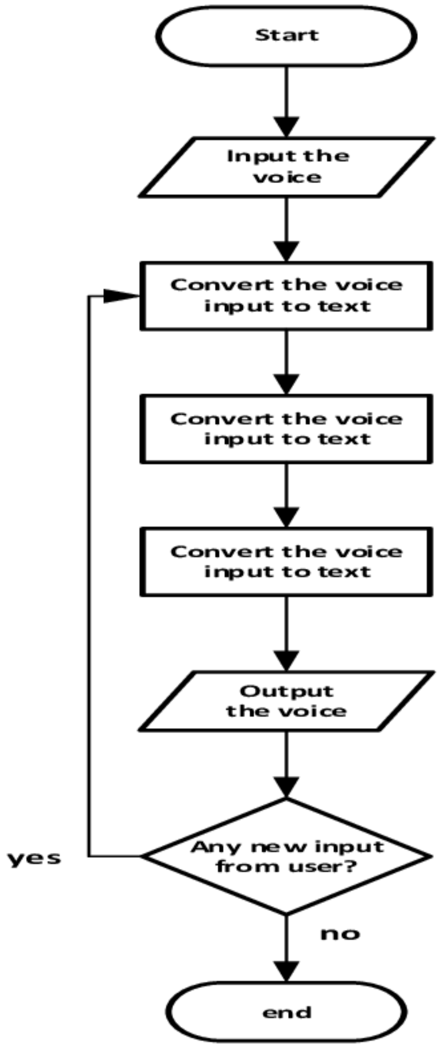
LOGIN
Login
Retrieve
User Login Db
Authenticate
USER
Ask Query
PROCESS QUERY
Search Fetched Keyword
Get Result
Output the query
Query Db
UPDATE LOGS
Store Update
Login Successful
ADMIN
Login
LOGIN
Retrieve
Admin Login Db
Authenticate

<u>Use Case Diagram:-</u> (User and admin roles)

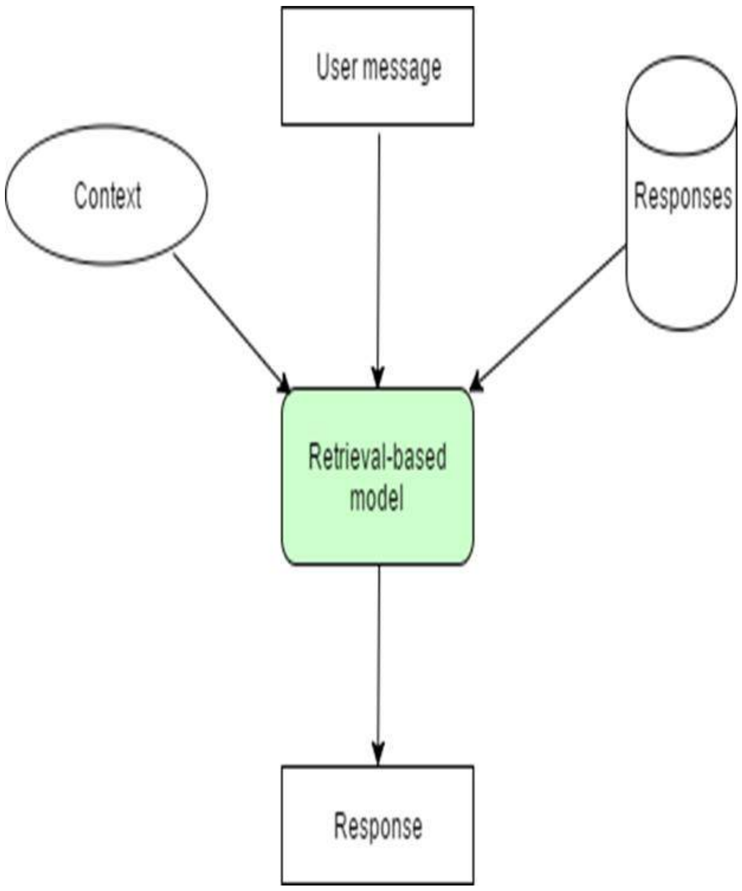
Fig 1: Use Case Diagram of user and admin roles.



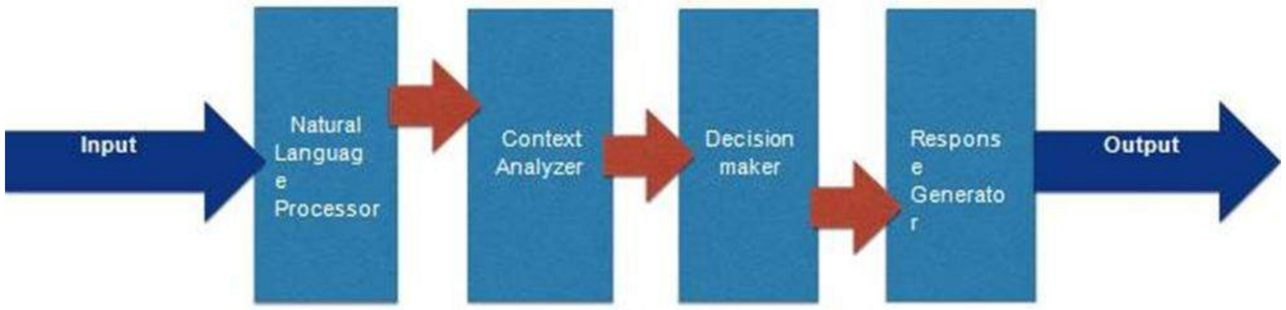
FLOW CHART DIAGRAM:-


Start
Input the voice
Convert the voice input to text
Convert the voice input to text
Convert the voice input to text
Output the voice
Any new input from user?
yes
no
end

ARCHITECTURE DIAGRAM:-



BASIC COMPONENTS OF CHATBOT:-



2. Literature Review

BANK MANAGEMENT SYSTEM tackle as a project is based on applicable technologies. The main aim of this project is to enlarge software for bank management system. This project is to enlarge software for bank management system. This project has been enlarge to carry out the things easily and quickly, which is not feasible with the manuals systems, which are control by this software. This project will be developed using Python language and. Hence it provides the proper solution for the present management system.

2.1 Natural Language Processing

Natural Language Processing (NLP) is the study of letting computers understand human languages. Without NLP, human language sentences are just a series of meaningless symbols to computers. Computers don’t recognize the words and don’t understand the grammars. NLP can be regard as a “translator”, who will translate human languages to computer understandable information. Traditionally, users need to follow well-defined procedures accurately, in order to interact with computers. For example, in Linux systems, all commands must be precise. Apple Siri and Microsoft Cortana have made it possible to give command in everyday langu ages and is changing the way of interacting.

2.2 Machine Learning.

Machine Learning (ML) is an area of computer science that "gives computers the ability to learn without being explicitly programmed". The parameter of the formulas is calculated from the data, rather than defined by the programmer. Two most common usage of ML is Classification and Regression. As shown in figure1, Classification means to categorize different types of data, while Regression means to find a way to describe the data. Basic ML program will have two stages, ***fitting*** and ***predicting.*** In the fitting stage, the program will be given a large set (at least thousands) of data. The program will try to adjust its parameter based on some statistical models, in order to make it “fit” the input data best. In the predicting stage, the program will givea prediction for a new input based on the parameters it just calculated out. For example, the famous Iris flower dataset contains the measurement of several features of three different species of flowers, such as the length of sepals and petals. A well-defined ML program can learn the pattern behind this feature and give prediction accordingly.

The term machine learning was coined in 1959 by Arthur Samuel, an American IBMer and pioneer in the field of computer gaming and artificial intelligence. A representative book of the machine learning research during the 1960s was the Nilsson's book on Learning Machines, dealing mostly with machine learning for pattern classification. Interest related to pattern recognition continued into the 1970s, as described by Duda and Hart in 1973. In 1981 a report was given on using teaching strategies so that a neural network learns to recognize 40 characters (26 letters, 10 digits, and 4 special symbols) from a computer terminal.

Tom M. Mitchell provided a widely quoted, more formal definition of the algorithms studied in the machine learning field: "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E." This definition of the tasks in which machine learning is concerned offers a fundamentally operational definition rather than defining the field in cognitive terms. This follows Alan Turing's proposal in his paper "Computing Machinery and Intelligence", in which the question "Can machines think?" is replaced with the question "Can machines do what we (as thinking entities) can do?".

Modern day machine learning has two objectives, one is to classify data based on models which have been developed, the other purpose is to make predictions for future outcomes based on these models. A hypothetical algorithm specific to classifying data may use computer vision of moles coupled with supervised learning in order to train it to classify the cancerous moles. Whereas, a machine learning algorithm for stock trading may inform the trader of future potential predictions.

CHAPTER-3(PROBLEM FORMULATION)

Problem formulation:-

The main difficulty that makes an appearance during the project emergence will be carry on the users of the bank. If a user gives up then all its data should also be detached so that we can direct the data dimensions also. And at the time of adding a new user the same type of enterprise should be lay hold of. The bank management system is an application for keep going a person's account in a bank. The system supplies the access to the customer to create an account, deposit /withdraw the cash from his account, also to view details of all accounts currently available. The following presentation comes up with the identification for the arrangement.

• **Lacking of training data.** The quantity and quality of training data is critical to the performance to a machine learning model. However, because some confidential and privacy reasons, the business team cannot provide enough data for us, and we had to make up data by our own. For the machine learning model, we generate some fake data based on our daily life experience, which is really biased, although with a good accuracy on the fake data.

• **Unstable API version.** Because API services we are using are still underdevelopment, and we cannot fix to a version for the API, the API may changes overtime. Moreover, there are inconsistencies between the APIs and their documents or sample codes.

CHAPTER – 4 (SYSTEM REQUIREMENTS)

4.1 Hardware Requirements:

Processor : Pentium IV(minimum)

Hard Disk : 40GB

RAM : 256MB (minimum)

4.2 Software Requirements:

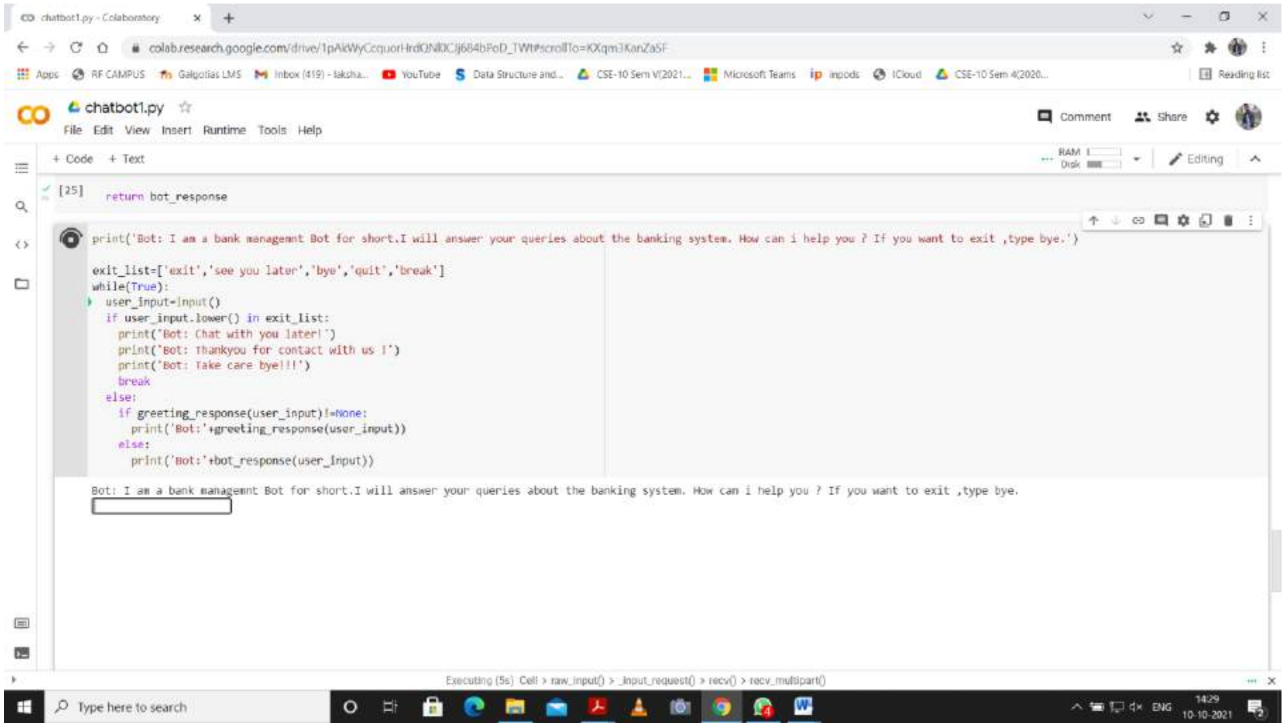
Operating System : Windows or Linux

Technology : PYTHON

CHAPTER-5(PROPOSED SOLUTION)

5.1 AIML Scripting

We created the AIML file that only handles one pattern, load aiml b. When we enter that command to the bot, it will try to load basic_chat.aiml. It won't work unless we actually create it. Here is what you can put inside basic_chat.aiml. We will match two basic patterns and respond.



```
[25]  return bot_response
```

```
print('Bot: I am a bank managemnt Bot for short.I will answer your queries about the banking system. How can i help you ? If you want to exit ,type bye.')

exit_list=['exit','see you later','bye','quit','break']
while(True):
  user_input=input()
  if user_input.lower() in exit_list:
    print('Bot: Chat with you later!')
    print('Bot: Thankyou for contact with us !')
    print('Bot: Take care bye!!!')
    break
  else:
    if greeting_response(user_input)!=None:
      print('Bot:'+greeting_response(user_input))
    else:
      print('Bot:'+bot_response(user_input))
```

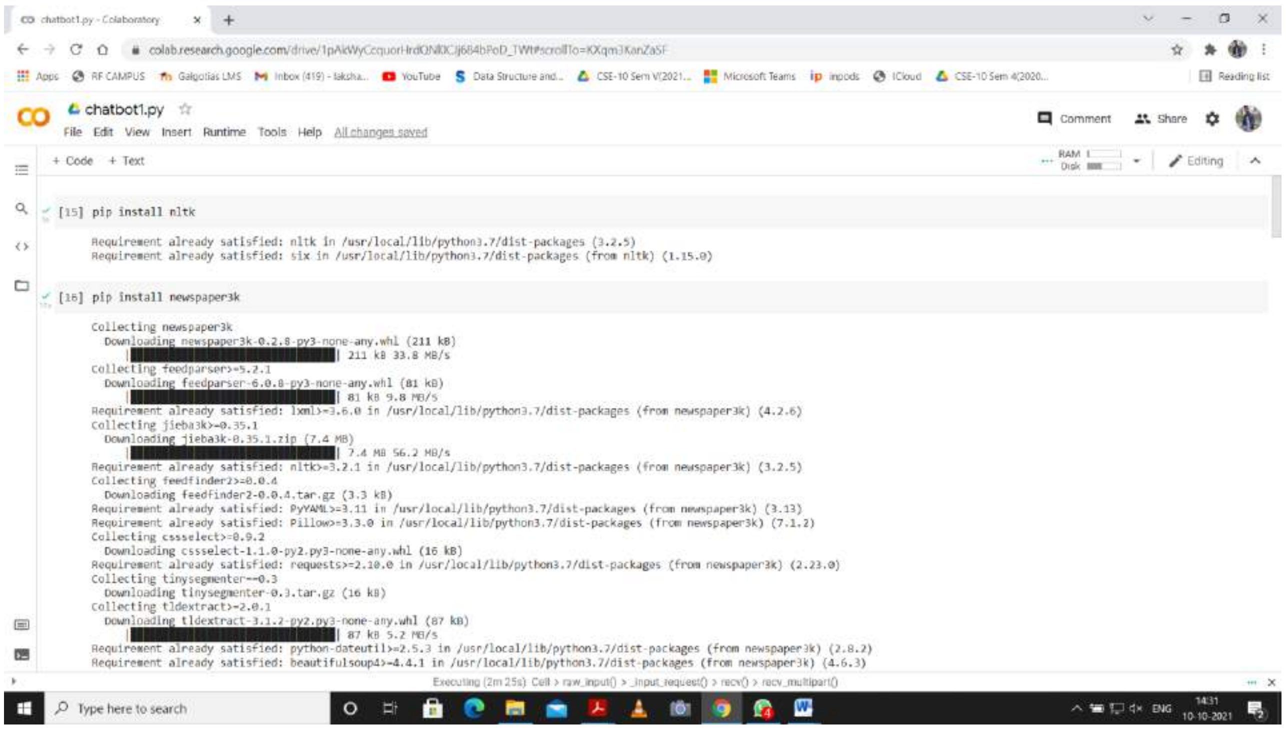
```
Bot: I am a bank managemnt Bot for short.I will answer your queries about the banking system. How can i help you ? If you want to exit ,type bye.
```

Executing (5s) Cell > raw_input() > _input_request() > recv() > recv_multipart()

5.2 Creating Interface-

As shown in the figure, there are six components in our project. The *interfaces* are the front end chat box for user to talk to the bot, which can be the Bot Portal, Skype, Facebook, etc. The ***connector*** works as a common gateway for all the interfaces. The outbound side calls different APIs to different front end, but the inbound APIs kept the same for our bot to connect. Fortunately, this connector has already been implemented by the bot framework SDK, we only need to rightly configure them.

The ***bot*** part contains the main flow control of our project. It is responsible for redirect the input to different models, parse the return values, and determines what to do next. It is also connected to the database to retrieve and update values.



```
[15] pip install nltk

Requirement already satisfied: nltk in /usr/local/lib/python3.7/dist-packages (3.2.5)
Requirement already satisfied: six in /usr/local/lib/python3.7/dist-packages (from nltk) (1.15.0)
```

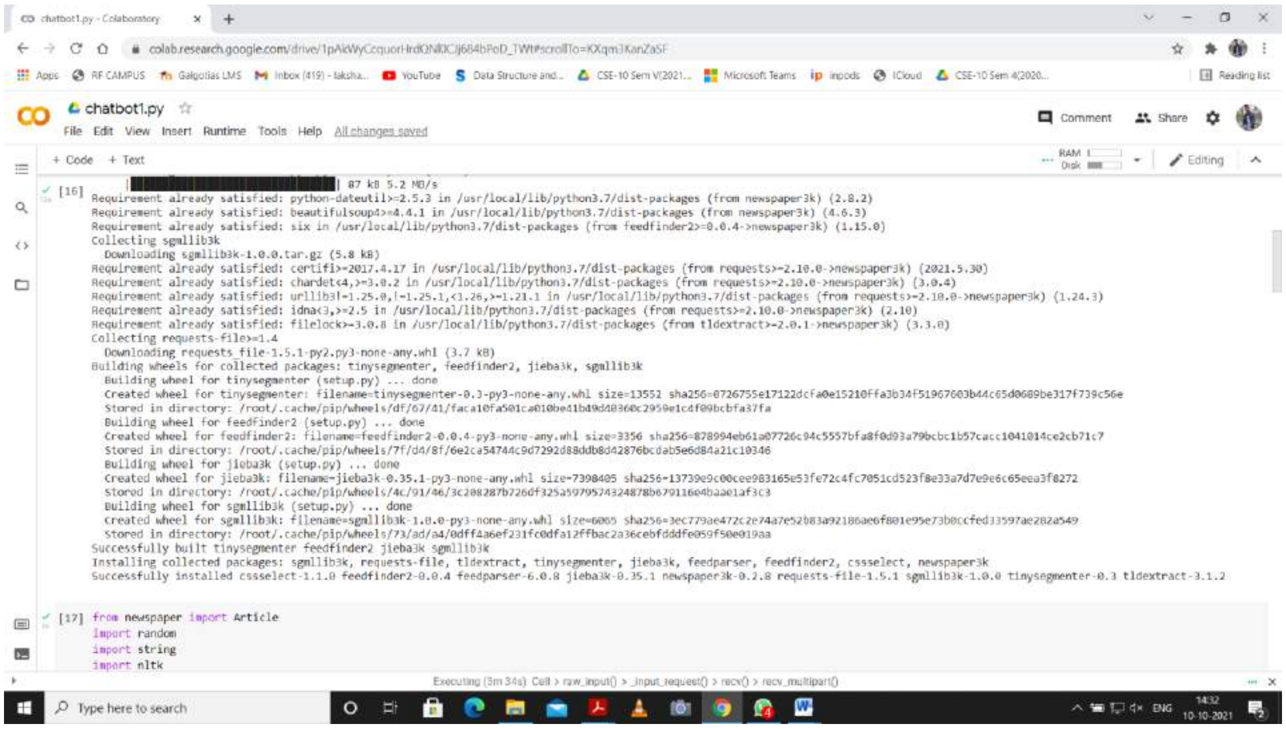
```
[16] pip install newspaper3k

Collecting newspaper3k
  Downloading newspaper3k-0.2.8-py3-none-any.whl (211 kB)
     |████████████████████████████████| 211 kB 33.8 MB/s
Collecting feedparser>=5.2.1
  Downloading feedparser-6.0.8-py3-none-any.whl (81 kB)
     |████████████████████████████████| 81 kB 9.8 MB/s
Requirement already satisfied: lxml>=3.6.0 in /usr/local/lib/python3.7/dist-packages (from newspaper3k) (4.2.6)
Collecting jieba3k>=0.35.1
  Downloading jieba3k-0.35.1.zip (7.4 MB)
     |████████████████████████████████| 7.4 MB 56.2 MB/s
Requirement already satisfied: nltk>=3.2.1 in /usr/local/lib/python3.7/dist-packages (from newspaper3k) (3.2.5)
Collecting feedfinder2>=0.0.4
  Downloading feedfinder2-0.0.4.tar.gz (3.3 kB)
Requirement already satisfied: PyYAML>=3.11 in /usr/local/lib/python3.7/dist-packages (from newspaper3k) (3.13)
Requirement already satisfied: Pillow>=3.3.0 in /usr/local/lib/python3.7/dist-packages (from newspaper3k) (7.1.2)
Collecting cssselect>=0.9.2
  Downloading cssselect-1.1.0-py2.py3-none-any.whl (16 kB)
Requirement already satisfied: requests>=2.10.0 in /usr/local/lib/python3.7/dist-packages (from newspaper3k) (2.23.0)
Collecting tinysegmenter==0.3
  Downloading tinysegmenter-0.3.tar.gz (16 kB)
Collecting tldextract>=2.0.1
  Downloading tldextract-3.1.2-py2.py3-none-any.whl (87 kB)
     |████████████████████████████████| 87 kB 5.2 MB/s
Requirement already satisfied: python-dateutil>=2.5.3 in /usr/local/lib/python3.7/dist-packages (from newspaper3k) (2.8.2)
Requirement already satisfied: beautifulsoup4>=4.4.1 in /usr/local/lib/python3.7/dist-packages (from newspaper3k) (4.6.3)
```

chatbot1.py
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
Executing (2m 25s) Cell > raw_input() > _input_request() > recv() > recv_multipart()

CHAPTER-6(IMPLEMENTATION)

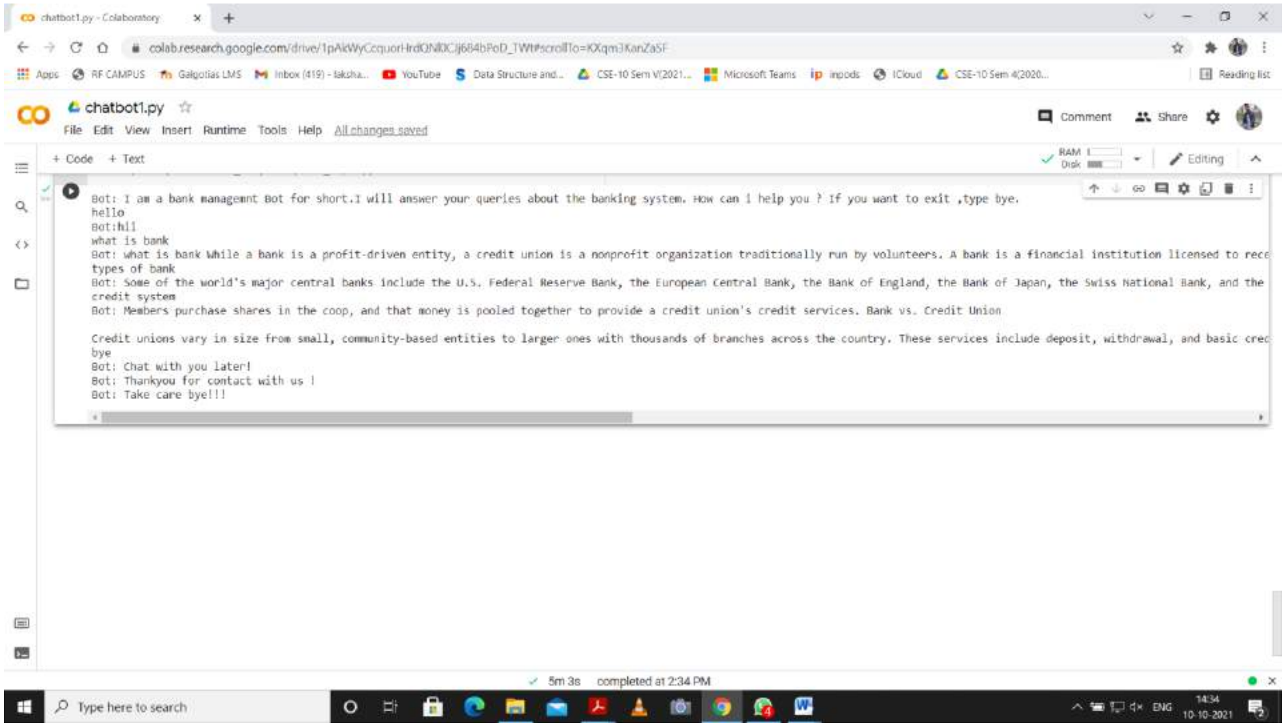
Implementation

This section covers the design and implementation of different module of the bot, which contains the design of the PYTHON module, the Translator API and the AIML module.



```
     |████████████████████████████████| 87 kB 5.2 MB/s
[16] Requirement already satisfied: python-dateutil>=2.5.3 in /usr/local/lib/python3.7/dist-packages (from newspaper3k) (2.8.2)
Requirement already satisfied: beautifulsoup4>=4.4.1 in /usr/local/lib/python3.7/dist-packages (from newspaper3k) (4.6.3)
Requirement already satisfied: six in /usr/local/lib/python3.7/dist-packages (from feedfinder2>=0.0.4->newspaper3k) (1.15.0)
Collecting sgmllib3k
  Downloading sgmllib3k-1.0.0.tar.gz (5.8 kB)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.7/dist-packages (from requests>=2.10.0->newspaper3k) (2021.5.30)
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.7/dist-packages (from requests>=2.10.0->newspaper3k) (3.0.4)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /usr/local/lib/python3.7/dist-packages (from requests>=2.10.0->newspaper3k) (1.24.3)
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-packages (from requests>=2.10.0->newspaper3k) (2.10)
Requirement already satisfied: filelock>=3.0.8 in /usr/local/lib/python3.7/dist-packages (from tldextract>=2.0.1->newspaper3k) (3.3.0)
Collecting requests-file>=1.4
  Downloading requests_file-1.5.1-py2.py3-none-any.whl (3.7 kB)
Building wheels for collected packages: tinysegmenter, feedfinder2, jieba3k, sgmllib3k
  Building wheel for tinysegmenter (setup.py) ... done
  Created wheel for tinysegmenter: filename=tinysegmenter-0.3-py3-none-any.whl size=13552 sha256=0726755e17122dcfa0e15210ffa3b34f51967603b44c65d0689be317f739c56e
  Stored in directory: /root/.cache/pip/wheels/df/67/41/faca10fa501ca010be41b49d40360c2959e1cdf09bcbfa37fa
  Building wheel for feedfinder2 (setup.py) ... done
  Created wheel for feedfinder2: filename=feedfinder2-0.0.4-py3-none-any.whl size=3356 sha256=878994eb61a07726c94c5557bfa8f0d93a79bcbc1b57cacc1041014ce2cb71c7
  Stored in directory: /root/.cache/pip/wheels/7f/d4/8f/6e2ca54744c9d7292d88ddb8d42876bcdab5e6d84a21c10346
  Building wheel for jieba3k (setup.py) ... done
  Created wheel for jieba3k: filename=jieba3k-0.35.1-py3-none-any.whl size=7398405 sha256=13739e9c00cee983165e53fe72c4fc7051cd523f8e33a7d7e9e6c65eea3f8272
  Stored in directory: /root/.cache/pip/wheels/4c/91/46/3c208287b726df325a597957432487 8b679116e4baae1af3c3
  Building wheel for sgmllib3k (setup.py) ... done
  Created wheel for sgmllib3k: filename=sgmllib3k-1.0.0-py3-none-any.whl size=6065 sha256=3ec779ae472c2e74a7e52b83a92186ae6f801e95e73b0ccfed33597ae282a549
  Stored in directory: /root/.cache/pip/wheels/73/ad/a4/0dff4a6ef231fc0dfa12ffbac2a36cebfdddfe059f50e019aa
Successfully built tinysegmenter feedfinder2 jieba3k sgmllib3k
Installing collected packages: sgmllib3k, requests-file, tldextract, tinysegmenter, jieba3k, feedparser, feedfinder2, cssselect, newspaper3k
Successfully installed cssselect-1.1.0 feedfinder2-0.0.4 feedparser-6.0.8 jieba3k-0.35.1 newspaper3k-0.2.8 requests-file-1.5.1 sgmllib3k-1.0.0 tinysegmenter-0.3 tldextract-3.1.2

[17] from newspaper import Article
     import random
     import string
     import nltk
```

CHATBOT INTERFACE:-



```
Bot: I am a bank managemnt Bot for short.I will answer your queries about the banking system. How can i help you ? If you want to exit ,type bye.
hello
Bot:hi!
what is bank
Bot: what is bank While a bank is a profit-driven entity, a credit union is a nonprofit organization traditionally run by volunteers. A bank is a financial institution licensed to rece
types of bank
Bot: Some of the world's major central banks include the U.S. Federal Reserve Bank, the European Central Bank, the Bank of England, the Bank of Japan, the Swiss National Bank, and the
credit system
Bot: Members purchase shares in the coop, and that money is pooled together to provide a credit union's credit services. Bank vs. Credit Union

Credit unions vary in size from small, community-based entities to larger ones with thousands of branches across the country. These services include deposit, withdrawal, and basic crec
bye
Bot: Chat with you later!
Bot: Thankyou for contact with us !
Bot: Take care bye!!!
```

CHAPTER – 7 (ADVANTAGES AND DISADVANTAGES)

7.1 Advantages

1. Accessible anytime:

I'm sure most of you are always kept on hold while operators connect you to a customer care executive. On an average people spend around 7 minutes until they are assigned to a person. Gone are the frustrating days of waiting in a queue for the next available operative. They are replacing live chat and other forms of slower contact methods such as emails and phone calls. Since chatbots are basically virtual robots they never get tired and continue to obey your command. They will continue to operate every day throughout the year without requiring to take a break. This improves your customer UX and helps you rank highly in your sector. Another advantage of this instant response is that you can also skillfully craft your chatbot to maintain your image and brand.

2. Handling Capacity:

Unlike humans who can only communicate with one human at a time, chat bots can simultaneously have conversations with thousands of people. No matter what time of the day it is or how many people are contacting you, every single one of them will be answered immediately.
Imagine you own a restaurant, and you have a good reputation for your food of which most of your revenues come from delivery. As the demand keeps rising, you will have more customers to take orders from but very few staff to attend them all. Having a chatbot would eliminate such problem and cater to each and every person and ensure that no order is missed. Companies like Taco Bell and Dominos are already using chatbots to arrange delivery of parcels.

3. Flexible attribute:

Chatbots have the benefit that it can quite easily be used in any industry. Unlike other products where you have to do a lot of development and testing to change platforms, chatbots are relatively easy to switch. One has to just train the bot by giving the right conversation structure and flow to switch its current field or industry. Or if there is a lot of back and forth between two sections of the industry say customer support and sales, then you could have custom built presets which would already have the conversation flow and structure to carry out the interactions with the user.

4. Customer Satisfaction:

Humans are bound to change of emotions. Chatbots, on the other hand, are bound by some rules and obey them as long as they're programmed to. They will always treat a customer in the perfect way no matter how rough the person is or how foul language the person uses.

Not everyone orders the same food everyday, people's choices may change everyday. In this case, it can use your order history to make suggestions for the next order, learn your address details and much more. Customers love this smooth interaction and want all their transactions to be as simple as possible.

5. Cost Effective:

Hiring a human for a job is never a cheap affair, and it will be expensive if your revenue are not high or sales targets are not met and would create havoc in the business. Due to the boundaries of human beings, a single human can only handle one or two people at the same time. More than that would be extremely tough for the employee.

Chatbots could help solve this age-old problem. As one chatbot is equal to loads of employees, it can easily communicate with thousands of customers at the same time. We would only need a handful of people to jump into conversations sometimes when necessary. Hence, it would drastically bring down the expenses and bring about a steep rise in revenue and customer satisfaction.

6. Faster Onboarding:

Before you want to accomplish a task, you first must learn how to work on the task and complete it. Only then will they be considered fit for the job. There is a continuous teaching involved in every level of hierarchy the employee will go through. Also, there will be a lot of change in the employees, some stay, some get fired, some more join in etc.

What we want to say is, employees will change; it's a fact. And this would require you to allot a lot time of your employees into grooming the new joinees. Chatbots could eliminate that time to almost zero, but provide a very clean and easy to understand conversation flow and structure that needs to be maintained by the chatbot. No doubt there will be changes in this too, but it will rather take a fraction of your time to resolve as compared to human employees.

7.2 Disadvantages-

1. Too many functions

Most of developers strive to create a universal chatbot that will become a fully-fledged assistant to user. But in practice functional bots turn out not to cope with the majority of queries. They often do not understand the user, forget what they were told 5 minutes earlier, and have many other disadvantages. And that is no wonder, as the development of a universal bot, which would understand natural language and could evaluate context, takes years of hard work for a team of experienced programmers. And even in this case, such programs should be constantly improved while in service. However, modern technologies allow building rather useful bots to perform specific actions such as booking train tickets or providing support to bank customers.
There are situations that depend on personal service. There are cases in which personal contact is irreplaceable in order to solve a problem. In addition, listening to what customers have to say is a matter of great importance for brands that want to develop and improve their customer service. Many customers prefer contact with real employees. Even companies that offer a variety of customer service options end up realizing that some of them still prefer to use more traditional methods of communication. Therefore, it is up to each company to carry out surveys to get to know its audience better and try to understand their degree of familiarity with the technology.

2. Primitive algorithms

There are two types of bots: based on artificial intelligence, being able to learn in the process of communication; programmed for specific behavior scenarios. Artificial intelligence chatbots are considered to be better, as they can respond depending on the situation and context. But the development of complex algorithms is required for this purpose. Meanwhile, only IT giants and few developers possess such powerful technological base. Therefore, it would be better for ordinary companies to focus on the second variant of bots, as they are more reliable and simpler. Namely for the reason they do not possess intelligence, they will not be able to adopt rude communication patterns and get beyond the control of creators.

3. Complex interface

Talking to a bot implies talking in a chat, meaning that a user will have to write a lot. And in case a bot cannot understand the user's request, he will have to write even more. It takes time to find out which commands a bot can respond to correctly, and which questions are better to avoid. Thus, talking to a chatbot does not save time in the majority of cases. Perhaps the efficiency of virtual assistants will increase due to the implementation of voice recognition function in the future. But for the time being their functional capabilities are very restricted, and they can be truly useful only in a few business areas.

CHAPTER – 8 (TECHNOLOGIES)

ABOUT PYTHON:

Dating from 1991, Python is a relatively new programming language. From the start, Python was considered a gap-filler, a way to write scripts that “automate the boring stuff” (as one popular book on learning Python put it) or to rapidly prototype applications that will be implemented in one or more other languages.

However, over the past few years, Python has emerged as a first-class citizen in modern software development, infrastructure management, and data analysis. It is no longer a back- room utility language, but a major force in web application development and systems management and a key driver behind the explosion in big data analytics and machine

Perfect for IT, Python simplifies many kinds of work, from system automation to working in cutting-edge fields like machine learning.

Python is easy to learn. The number of features in the language itself is modest, requiring relatively little investment of time or effort to produce one’s first programs. Python syntax is designed to be readable and straightforward. This simplicity makes Python an ideal teaching language, and allows newcomers to pick it up quickly. Developers spend more time thinking about the problem they’re trying to solve, and less time thinking about language complexities or deciphering code left by others.

Python is broadly used and supported. Python is both popular and widely used, as the high rankings in surveys like the Tiobe Index and the large number of GitHub projects using Python attest. Python runs on every major operating system and platform, and most minor ones too. Many major libraries and API-powered services have Python bindings or wrappers, allowing Python to interface freely with those services or make direct use of those libraries. Python may not be the fastest language, but what it lacks in speed, it makes up for in versatility.

Python is not a “toy” language. Even though scripting and automation cover a large chunk of Python’s use cases (more on that below), Python is also used to build robust, professional- quality software, both as standalone applications and as web services.

Rather than having all of its functionality built into its core, Python was designed to be highly extensible (with modules). This compact modularity has made it particularly popular as a means of adding programmable interfaces to existing applications.

Van Rossum's vision of a small core language with a large standard library and easily extensible interpreter stemmed from his frustrations with ABC, which espoused the opposite approach.
Python strives for a simpler, less-cluttered syntax and grammar while giving developers a choice in their coding methodology. In contrast to Perl's "there is more than one way to do it" motto, Python embraces a "there should be one— and preferably only one —obvious way to do it" design philosophy. Alex Martelli, a Fellow at the Python Software Foundation and Python book author, writes that "To describe something as 'clever' is not considered a compliment in the Python culture."
Python's developers strive to avoid premature optimization, and reject patches to non-critical parts of the CPython reference implementation that would offer marginal increases in speed at the cost of clarity. When speed is important, a Python programmer can move time-critical functions to extension modules written in languages such as C, or use PyPy, a just-in-time compiler. Cython is also available, which translates a Python script into C and makes direct C-level API calls into the Python interpreter.
An important goal of Python's developers is keeping it fun to use. This is reflected in the language's name—a tribute to the British comedy group Monty Python—and in occasionally playful approaches to tutorials and reference materials, such as examples that refer to spam and eggs (from a famous Monty Python sketch) instead of the standard foo and bar.
A common neologism in the Python community is pythonic, which can have a wide range of meanings related to program style. To say that code is pythonic is to say that it uses Python idioms well, that it is natural or shows fluency in the language, that it conforms with Python's minimalist philosophy and emphasis on readability.

What is Python used for?

The most basic use case for Python is as a scripting and automation language. Python isn’t just a replacement for shell scripts or batch files, but is also used to automate interactions with web browsers or application GUIs or system provisioning and configuration in tools such as Ansible and Salt. But scripting and automation represent only the tip of the iceberg with Python.

- Python is used for general application programming. Both CLI and cross-platform GUI applications can be created with Python and deployed as self-contained executables. Python doesn’t have the native ability to generate a standalone binary from a script, but third-party packages like cx_Freeze or PyInstaller can be used to accomplish that.

- Python is used for data science and machine learning. Sophisticated data analysis has become one of fastest moving areas of IT and one of Python's star use cases. The vast majority of the libraries used for data science or machine learning have Python interfaces, making the language the most popular high-level command interface to for machine learning libraries and other numerical algorithms.
- Python is used for web services and RESTful APIs. Python's native libraries and third-party web frameworks provide fast and convenient ways to create everything from simple REST APIs in a few lines of code, to full-blown, data-driven sites. Python's latest versions have powerful support for asynchronous operations, allowing sites to handle up to tens of thousands of requests per second with the right libraries.
- Python is used for metaprogramming. In Python, everything in the language is an object, including Python modules and libraries themselves. This allows Python to work as a highly efficient code generator, making it possible to write applications that manipulate their own functions and have the kind of extensibility that would be difficult or impossible to pull off in other languages.
- Python is used for glue code. Python is often described as a "glue language," meaning it can allow disparate code (typically libraries with C language interfaces) to interoperate. Its use in data science and machine learning is in this vein, but that's just one incarnation of the general idea.

Also worth noting are the sorts of tasks Python is not well-suited for. Python is a high-level language, so it's not suitable for system-level programming—device drivers or OS kernels are straight out. It's also not ideal for situations that call for cross-platform standalone binaries. You could build a standalone Python app for Windows, Mac, and Linux, but not elegantly or simply. Finally, Python is not the best choice when speed is an absolute priority in every aspect of the application. For that you're better off with C/C++ or another language of that caliber.

The Python language's pros and cons

Python syntax is meant to be readable and clean, with little pretense. A standard "hello world" in Python 3.x is nothing more than:

- print("Hello world!")
- Python provides many syntactical elements that make it possible to concisely express many common program flows. Consider a sample program for reading lines from a text file into a list object, stripping each line of its terminating newline character along the way:
- with open('myfile.txt') as my_file:
- file_lines = [x.strip('\n') for x in my_file]
- The with/as construction is a "context manager," which provides an efficient way to instantiate a given object for a block of code and then dispose of it outside of that block. In this case, the object in question is my_file, instantiated with the open() function. This takes the place of several lines of boilerplate to open the file, read individual lines from it, then close it up.
- The [x … for x in my_file] construction is another Python idiosyncrasy, the "list comprehension." It allows a given item that contains other items (here, my_file and the lines it contains) to be iterated through, and to allow each iterated element (that is, each x) to be processed and automatically appended into a list.
- You could write such a thing as a formal for… loop in Python, much as you would in another language. The point is that Python has a way to economically express things like loops that iterate over multiple objects and perform some simple operation on each element in the loop, or work with things that require explicit instantiation and disposal. Constructions like this allow Python developers to balance terseness and readability.
- Python's other language features are meant to complement common use cases. Most modern object types—Unicode strings, for instance—are built directly into the language. Data structures—like lists, dictionaries (i.e., hashmaps), tuples (for storing immutable collections of objects), and sets (for storing collections of unique objects)—are available as standard-issue items.

- Like C#, Java, and Go, Python has garbage-collected memory management, meaning the programmer doesn’t have to implement code to track and release objects. Normally garbage collection happens automatically in the background, but if that poses a performance problem, it can be triggered manually or disabled entirely.
- An important aspect of Python is its dynamism. Everything in the language, including functions and modules themselves, are handled as objects. This comes at the expense of speed (more on that below), but makes it far easier to write high-level code. Developers can perform complex object manipulations with only a few instructions, and even treat parts of an application as abstractions that can be altered if needed.
- Python’s use of significant whitespace has been cited as both one of Python’s best and worst attributes. The indentation on the second line shown above isn’t just for readability; it is part of Python’s syntax. Python interpreters will reject programs that don’t use proper indentation to indicate control flow.
- Syntactical white space might cause noses to wrinkle, and some people do reject Python out of hand for this reason. But strict indentation rules are far less obtrusive in practice than they might seem in theory, even with the most minimal of code editors, and the end result is code that is cleaner and more readable.

Python 2 versus Python 3

- Python is available in two versions, which are different enough to trip up many new users. Python 2.x, the older “legacy” branch, will continue to be supported (i.e. receive official updates) through 2020, and it might even persist unofficially after that. Python 3.x, the current and future incarnation of the language, has many useful and important features not found in 2.x, such as better concurrency controls and a more efficient interpreter.
- Python 3 adoption was slowed for the longest time by the relative lack of third-party library support. Many Python libraries supported only Python 2, making it difficult to switch. But over the last couple of years, the number of libraries supporting only Python 2 has dwindled; most are now compatible with both versions. Today, there are few reasons against using Python 3.

CHAPTER – 9 (CONCLUSION AND FUTURE SCOPE)

9.1 Conclusions

Chatbots are the new Apps! As we have discussed in the above deliverables, this project brings the power of chatbots to Yioop and enriches its usability. Chatbots in Yioop can give a human like touch to some aspects and make it an enjoying conversation. And they are focused entirely on providing information and completing tasks for the humans they interact with. The above mentioned functionality in all the deliverables is implemented and pushed in to Yioop code. By implementing the above mentioned deliverables I was able to add a basic chatbot functionality in to the Yioop. I.e., configuring and creating accounts for bot users with bot settings which is mentioned in deliverable 2, activating a bot whenever a user asks for it via post in a thread which is discussed in deliverable 3 and as I discussed in deliverable 4, I have implemented a simple weather chatbot that gives weather information whenever a user ask and Fig. 3 tells that I was also able to converse with the bot in Yioop. I intend to enhance the system developed so far in CS298. Next step towards building chatbots involve helping people to facilitate their work and interact with computers using natural language or using set of rules. Future Yioop chatbots, backed by machine-learning technology, will be able to remember past conversations and learn from them to answer new ones. The challenge would be conversing with multiple bot users and multiple users.

9.2 Future Scope

There are limitations to what has been currently achieved with chatbots. The limitations of data processing and retrieval are hindering chatbots to reach their full potential. It is not that we lack the computational processing power to do so. From gauging purchase intent to answering questions about IT issues, chatbots are on track to play a major role in the contemporary enterprise. Chatbots are fully functioning, semi-autonomous systems that can assist customer service experiences and response time. But that doesn't mean their future in the enterprise is secure. For chatbots to withstand the rapidly increasing technological shifts and become mainstays in the enterprise, developers need to examine the issues that have popped up with increased implementation.

The future scope of chatbots could include many benefits for enterprises, but experts say they will need to be gently nudged in the right direction for businesses to reap these benefits. However, there is a limitation on “How” we do it. One of the biggest examples is the retail customer market. Retail customers are primarily interested in interacting with humans because of nature of their needs. They don’t want bots to process their needs and respond accordingly.

References

1. Bayan Abu Shawar and Eric Atwell, 2007 “Chatbots: Are they Really Useful?”

2. LDV Forum - GLDV Journal for Computational Linguistics and Language Technology.

3. http://www.ldv-forum.org/2007_Heft1/Bayan_Abu-Shawar_and_Eric_Atwell.pdf

4. Bringing chatbots into education: Towards natural language negotiation of openlearner models. Know.-Based Syst. 20, 2 (Mar. 2007), 177-185.

5. Intelligent Tutoring Systems: Prospects for Guided Practice and Efficient Learning. Whitepaper for the Army's Science of Learning Workshop, Hampton, VA. Aug 1-3, 2006.

6. http://en.wikipedia.org/wiki/Chatterbot

7. ALICE. 2002. *A.L.I.C.E AI* Foundation, http://www.alicebot.org/