CHATBOT ON COLLEGE CONTENT RESPONSE USING NATURAL LANGUAGE PROCESSING

A PROJECT REPORT

Submitted by

GAYATHRI.C (411716104015)

SIVARANJANI.R (411716104042)

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

PRINCE SHRI VENKATESHWARA PADMAVATHY ENGINEERING COLLEGE, PONMAR, CHENNAI - 600 127

SEPTEMBER 2020

ANNA UNIVERSITY: CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report **“CHATBOT ON COLLEGE CONTENT RESPONSE USING NATURAL LANGUAGE PROCESSING”** is the bonafide work of **“GAYATHRI C(411716104015),SIVARANJANI R(411716104042)”** who carried out the project under my supervision.

Submitted for the project viva-voce examination held on __________

SIGNATURE	**SIGNATURE**
Mrs. R.REENA, M.E.	**Mrs. R.K.KAPILAVANI, M.E.,(PHD)**
HEAD OF THE DEPARTMENT	**SUPERVISOR**
ASSOCIATE PROFESSOR	**ASSISTANT PROFESSOR**
Department of CSE	Department of CSE
Prince Shri Venkateshwara	Prince Shri Venkateshwara Padmavathy
Padmavathy Engineering College,	Engineering College,
Ponmar,	Ponmar,
Chennai – 600 127	Chennai – 600 127

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

First and foremost, We wish to express our sincere thanks to our FOUNDER AND CHAIRMAN **Dr. K.VASUDEVAN, M.A., B.Ed., Ph.D.,** and VICE CHAIRMAN **Dr. V.VISHNU KARTHICK,MBBS., M.D,** for his endeavor in educating us in their premier institution.

We would like to express our deep gratitude to the ADMINISTRATIVE OFFICER, **Mr. K.PARTHASARATHY, B.E.,** for his valuable support.

We wish to express our appreciation and gracefulness to our PRINCIPAL, **Prof. MAHALAKSHMI.V, M.Tech., Ph.D.,** for his encouragement and sincere guidance.

We also wish to convey our thanks and gratitude to **Prof.R. REENA, M.E., HOD,** Department of Computer Science and Engineering, for her support and providing us ample time to complete our project.

We express our indebtedness to our project supervisor, **Prof.R.K.KAPILAVANI, M.E., Ph.D.,** Assistant Professor, Department of Information Technology , for her guidance throughout the course of our project and helping us to complete our project successfully.

We wish to convey our sincere thanks to all the teaching and non-teaching staff of Department of Computer Science and Engineering, without whose co-operation this venture would not have been success.

ABSTRACT

The development of chatbot has become trendier and so far several conversational chatbots were designed which replaces the traditional chatbots. A chatbot is a computer program which is used to interact with humans and fulfill their needs. Chatbot gives the response for the user query and sometimes they are capable of executing tasks also.

Early development of chatbots became so difficult whereas recent chatbots development is much easier because of the wide availability of development platforms and source code. A chatbot can be developed using either Natural Language Processing (NLP) or Deep Learning. When compared to traditional chatbots, bots designed using Deep Learning requires huge amount of data to train. A chatbot is a service, powered by rules and sometimes artificial intelligence, that you interact with via a chat interface. A chatbot can be deployed on various platforms such as mobile apps, web apps, messaging apps, personal assistant and what not. It’s job to provide human like interaction and assistance to the user. The best chatbot in the world would be one in which the user would be one in which the user would not be able to differentiate it from an actual human being. Natural Language Processing is an area of research and application that explores how computer can be used to understand and manipulate

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATION:

S.NO	ABBREVIATION	EXPANSION
01	ML	Machine Learning
02	AIML	Artificial Inteligence Machine Learning
03	NLP	Natural Language Processing
04	AI	Atificial Intelligence
05	UNIBOT	University Bot
06	SE4COG	Software Engineering for Cognitive services

CHAPTER 1

INTRODUCTION

1.1 Natural language processing (NLP)

It is a subfield of linguistics, computer science, information engineering, and artificial intelligence concerned with the interactions between computers and human (natural) languages, in particular how to program computers to process and analyze large amounts of natural language data

NLP-based bot entails :

1. Lesser false positive outcomes through accurate interpretation.

2. Identify user input failures and resolve conflicts using statistical modeling.

3. Use comprehensive communication for user responses.

4. Learn faster to address the development gaps.

5. Achieve natural language capability through lesser training data inputs.

6. Ability to re-purpose the input training data for future leanings.

7. Provide simple corrective actions for false positives.

This operation is divided into three parts:

• User post the query on chatbot

• Processing is done on the users query to match the predefined format by the developer

• Pattern matching is performed between user entered query and knowledge (pattern).

1.2 Chatbot

A chatbot is a service, powered by rules and sometimes artificial intelligence, that you interact with via a chat interface. A chatbot can be deployed on various platforms such as mobile apps, web apps, messaging apps, personal assistant and what not. It's job to provide human like interaction and assistance to the user. The best chatbot in the world would be one in which the user would not be able to differentiate it from an actual human being. Natural Language Processing is an area of research and application that explores how computer can be used to understand and manipulate Natural Language text or speech to do useful things.NLP is multi-disciplinary, it is closely related to linguistics. It also has links to computer and information sciences, psychology, electric and electronic engineering. It is also related to Artificial Intelligence. Applications of NLP include a number of fields of studies such as machine translation, speech recognition, text processing and summarization and so on. A chatbot can interact with users in various formats such as text, speech and actions. Chatbot is one of the hottest topics out there. Tech giants like Microsoft, Amazon, Google, Facebook have been competing to roll out better chatbots everyday.

1.3 OBJECTIVE

Chatbot could effectively answers College related queries with an added advantage that it also provides personal information like grades, etc. With proper user authentication. So the user gets the desired information quickly without having to go through a series of web pages.

1.4 MOTIVATION

It is often impossible to get all the data on a single interface without the complications of going through multiple forms and windows. The college chat bot aims to remove this difficulty by providing a common and user-friendly interface to solve queries of college students and teachers. The purpose of a chat bot 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.

1.5 DOMAIN DESCRIPTION

Machine Learning

Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. Machine learning focuses on the development of computer programs that can access data and use it learn for themselves.The process of learning begins with observations or data, such as examples, direct experience, or instruction, in order to look for patterns in data and make better decisions in the future based on the examples that we provide. The primary aim is to allow the computers learn automatically without human intervention or assistance and adjust actions accordingly.

1.5.1 Machine Learning Methods

Machine learning algorithms are often categorized as supervised or unsupervised.

Supervised machine learning algorithms can apply what has been learned in the past to new data using labeled examples to predict future events. Starting from the analysis of a known training dataset, the learning algorithm produces an inferred function to make predictions about the output values. The system is able to provide targets for any new input after sufficient training. The learning algorithm can also compare its output with the correct, intended output and find errors in order to modify the model accordingly.

In contrast, **unsupervised machine learning algorithms** are used when the information used to train is neither classified nor labeled. Unsupervised learning studies how systems can infer a function to describe a hidden structure from unlabeled data. The system doesn't figure out the right output, but it explores the data and can draw inferences from datasets to describe hidden structures from unlabeled data.

Semi-supervised machine learning algorithms fall somewhere in between supervised and unsupervised learning, since they use both labeled and unlabeled data for training – typically a small amount of labeled data and a large amount of unlabeled data. The systems that use this method are able to considerably improve learning accuracy. Usually, semi-supervised learning is chosen when the acquired labeled data requires skilled and relevant resources in order to train it / learn from it. Otherwise, acquiring unlabeled data generally doesn't require additional resources.

Reinforcement machine learning algorithms is a learning method that interacts with its environment by producing actions and discovers errors or rewards. Trial and error search and delayed reward are the most relevant characteristics of reinforcement learning. This method allows machines and software agents to automatically determine the ideal behaviour within a specific context in order to maximize its performance. Simple reward feedback is required for the agent to learn which action is best; this is known as the reinforcement signal.

1.5.2 Advantages of Machine Learning

Machine Learning undoubtedly helps people to work more creatively and efficiently. Basically, you too can delegate quite complex or monotonous work to the computer through Machine Learning - starting with scanning, saving and filing paper documents such as invoices up to organizing and editing images.

In addition to these rather simple tasks, self-learning machines can also perform complex tasks. These include, for example, the recognition of error patterns. This is a major advantage, especially in areas such as the manufacturing industry: the industry relies on continuous and error-free production. While even experts often cannot be sure where and by which correlation a production error in a plant fleet arises, Machine Learning offers the possibility to identify the error early this saves downtimes and money. Self-learning programs are now also used in the medical field. In the future, after "consuming" huge amounts of data (medical publications, studies, etc.), apps will be able to warn a in case his doctor wants to prescribe a drug that he cannot tolerate. This "knowledge" also means that the app can propose alternative options which for example also take into account the genetic requirements of the respective patient.

1.5.3 Applications of Machine Learning

1. Virtual Personal Assistants

Siri, Alexa, Google Now are some of the popular examples of virtual personal assistants. As the name suggests, they assist in finding information, when asked over voice. All you need to do is activate them and ask “What is my schedule for today?”, “What are the flights from Germany to London”, or similar questions. For answering, your personal assistant looks out for the information, recalls your related queries, or send a command to other resources (like phone apps) to collect info. You can even instruct assistants for certain tasks like “Set an alarm for 6 AM next morning”, “Remind me to visit Visa Office day after tomorrow”. Machine learning is an important part of these personal assistants as they collect and refine the information on the basis of your previous involvement with them. Later, this set of data is utilized to render results that are tailored to your preferences.Virtual Assistants are integrated to a variety of platforms. For example: Smart Speakers: Amazon Echo and Google HomeSmartphones: Samsung Bixby on Samsung S8Mobile Apps: Google Allo

2. Predictions while Commuting

Traffic Predictions: We all have been using GPS navigation services. While we do that, our current locations and velocities are being saved at a central server for managing traffic. This data is then used to build a map of current traffic. While this helps in preventing the traffic and does congestion analysis, the underlying problem is that there are less number of cars that are equipped with GPS. Machine learning in such scenarios helps to estimate the regions where congestion can be found on the basis of daily experiences.

Online Transportation Networks: When booking a cab, the app estimates the price of the ride. When sharing these services, how do they minimize the detours? The answer is machine learning. Jeff Schneider, the engineering lead at Uber ATC reveals in a an interview that they use ML to define price surge hours by predicting the rider demand. In the entire cycle of the services, ML is playing a major role.

3. Videos Surveillance

Imagine a single person monitoring multiple video cameras! Certainly, a difficult job to do and boring as well. This is why the idea of training computers to do this job makes sense.The video surveillance system nowadays are powered by AI that makes it possible to detect crime before they happen. They track unusual behaviour of people like standing motionless for a long time, stumbling, or napping on benches etc. The system can thus give an alert to human attendants, which can ultimately help to avoid mishaps. And when such activities are reported and counted to be true, they help to improve the surveillance services. This happens with machine learning doing its job at the backend.

4. Social Media Services

From personalizing your news feed to better ads targeting, social media platforms are utilizing machine learning for their own and user benefits. Here are a few examples that you must be noticing, using, and loving in your social media accounts, without realizing that these wonderful features are nothing but the applications of ML.People You May Know: Machine learning works on a simple concept: understanding with experiences. Facebook continuously notices the friends that you connect with, the profiles that you visit very often, your interests, workplace, or a group that you share with someone etc. On the basis of continuous learning, a list of Facebook users are suggested that you can become friends with.Face Recognition: You upload a picture of you with a friend and Facebook instantly recognizes that friend. Facebook checks the poses and projections in the picture, notice the unique features, and then match them with the people in your friend list. The entire process at the backend is complicated and takes care of the precision factor but seems to be a simple application of ML at the front end.Similar Pins: Machine learning is the core element of Computer Vision, which is a technique to extract useful information from images and videos. Pinterest uses computer vision to identify the objects (or pins) in the images and recommend similar pins accordingly.

5. Email Spam and Malware Filtering

There are a number of spam filtering approaches that email clients use. To ascertain that these spam filters are continuously updated, they are powered by machine learning. When rule-based spam filtering is done, it fails to track the latest tricks adopted by spammers. Multi Layer Perception, C 4.5 Decision Tree

Induction are some of the spam filtering techniques that are powered by ML.Over 325, 000 malwares are detected everyday and each piece of code is 90–98% similar to its previous versions. The system security programs that are powered by machine learning understand the coding pattern. Therefore, they detects new malware with 2–10% variation easily and offer protection against them.

6. Online Customer Support

A number of websites nowadays offer the option to chat with customer support representative while they are navigating within the site. However, not every website has a live executive to answer your queries. In most of the cases, you talk to a chatbot. These bots tend to extract information from the website and present it to the customers. Meanwhile, the chatbots advances with time. They tend to understand the user queries better and serve them with better answers, which is possible due to its machine learning algorithms.

7. Search Engine Result Refining

Google and other search engines use machine learning to improve the search results for you. Every time you execute a search, the algorithms at the backend keep a watch at how you respond to the results. If you open the top results and stay on the web page for long, the search engine assumes that the the results it displayed were in accordance to the query. Similarly, if you reach the second or third page of the search results but do not open any of the results, the search engine estimates that the results served did not match requirement. This way, the algorithms working at the backend improve the search results.

8. Product Recommendations

You shopped for a product online few days back and then you keep receiving emails for shopping suggestions. If not this, then you might have noticed that the shopping website or the app recommends you some items that somehow matches with your taste. Certainly, this refines the shopping experience but did you know that it’s machine learning doing the magic for you? On the basis of your behaviour with the website/app, past purchases, items liked or added to cart, brand preferences etc., the product recommendations are made.

9. Online Fraud Detection

Machine learning is proving its potential to make cyberspace a secure place and tracking monetary frauds online is one of its examples. For example: Paypal is using ML for protection against money laundering. The company uses a set of tools that helps them to compare millions of transactions taking place and distinguish between legitimate or illegitimate transactions taking place between the buyers and sellers.

CHAPTER 2

LITERATURE SURVEY

2.1 Chatbots and conversational agents: A bibliometric analysis

Authors: H. N. Io, C. B. Lee, 2017

Chatbots are replacing some of the jobs that are traditionally performed by human workers, such as online customer service agents and educators. From the initial stage of rule-based chatbots to the era of rapid development in artificial intelligence (AI), the performance of chatbots keeps improving. Chatbots can nowadays "chat" like a human being and they can learn from experience. The purpose of this research is to examine the past research on chatbots (also known as conversational agents) using the quantitative bibliometric analysis. The contribution of this research is to help researchers to identify research gaps for the future research agenda in chatbots. The results of the analysis found a potential research opportunity in chatbots due to the emergence of the deep learning technology.

2.1.1 Drawbacks

- Tends to stimulate one-dimensional thinking.
- It requires skills to calculate and interpret results.

2.2 Intelligent Chatbot for Easy Web-Analytics Insights

Authors: Ramya Ravi, 2018

In this fast-moving data-driven world, it is vital that to draw the accurate insights to make the right decisions at the right time. In terms of online websites, there are many web analytics tools that will give us performance reports. It is tedious and time consuming to master the tools leave alone to derive insights to understand the business impacts. In this paper, I am comparing 2 widely used analytics tools based on their ease of use. An Artificial Intelligence Machine Learning (AIML) driven chatbot, that is fueled with analytics' raw data, that will enable bot-users to get business insights by just typing in the query.

2.2.1 Drawbacks

- Pose an existential risk. It can be wrong.
- Expensive to develop

2.3 AI and Web-Based Human-Like Interactive University Chatbot (UNIBOT) Authors:Neelkumar P. Patel , Devangi R. Parikh , Darshan A. Patel , Ronak R. Patel, 2019

Most of the time, Students have to visit universities or colleges to collect various information like Tution fees, Term Schedule, etc. during their admission process or as per their daily needs. It is very tedious and time consuming, also it requires manpower in providing required information to visitors. Hence, to overcome the problems a chatbot can be developed. The project is about interaction between users and chatbot which can be accessed from anywhere anytime. The chatbot can be easily attached with any university or college website with few simple language conversions. Chatbot provides various information related to university or college and also students-related information. The chatbot can be used by anyone who can access the university's website. The project uses the concept of Artificial Intelligence and Machine Learning.

2.3.1 Drawbacks

- High Costs of Creation
- Lacking Out of Box Thinking

2.4 Chatbot using Tensor Flow for small Businesses

Authors: Rupesh Singh,Manmath Paste, Nirmala Shinde, Harshkumar Patel, Nitin Mishra, 2018

Chatbots are software used in entertainment industry, business and user support. Chatbots are modeled on various techniques such as knowledge base, machine learning based. Machine learning based chatbots yields more practical results. Chatbot which gives responses based on the context of conversation tends to be more user friendly. It demonstrates a method of developing chatbot which can follow the context of the conversation. It uses TensorFlow for developing the neural network model of the chatbot and uses the nlp techniques to maintain the context of the conversation. This chatbots can be used in small industries or business for automating customer care as user queries will be handled by chatbots

2.4.1 Drawbacks

- Tensor Flow does not offer functionality
- lacks in both speed and usage
- No support for OpenCV

2.5 Enterprise Crowd Computing for Human Aided Chatbots

Author: Alessandro Bozzon

2018 IEEE/ACM 1st International Workshop on Software Engineering for Cognitive Services (SE4COG)

A chatbot is an example of cognitive computing system that emulates human conversations to provide informational, transactional, and conversational services. Despite their widespread adoption, chatbots still su er from a number of performance issue due to limitations with their programming and training. In this paper we discuss Human Aided Chatbots, i.e. chatbots that rely on humans in the loop to operate. Human Aided Chatbots exploit human intelligence, brought for instance by crowd workers or full-time employees, to fill the gaps caused by limitations of fully automated solutions. A recent example of Human Aided Chatbots is Facebook M. To achieve broader adoption, Human Aided Chatbots must overcome a number of issues, including scalability, low-latency, and privacy. In this short paper, we discuss how Crowd Computing performed in the enterprise could help overcoming such issues. It had present some recent findings in the field of Enterprise Crowd Computing, and introduce ECrowd, a platform for enterprise crowd computing designed for gathering training data for cognitive systems.

2.7.1 Drawbacks

- Poor processing which is not able to filter results in time that can annoy people.
- Inability to understand.
- Customer dissatisfaction

CHAPTER 3

SYSTEM ANALYSIS

3.1 Existing System

Creating a database Two dimensional string arrays are applied to build a database. Rows in the array are used for request and response. All the even rows contain the request or questions and all the odd rows contain the response or answers. Columns in the array are applied to save different types of questions that could asked by the user and responses that a Chatbot can answer. There is one row in the array which contains default responses which is used when the matching question is not found in the array.

The machine has been embedded knowledge to identify the sentences and making a decision itself as response to answer a question. The response principle is matching the input sentence from user. From input sentence, it will be scored to get the similarity of sentences, the higher score obtained the more similar of reference sentences. The sentence similarity calculation in this paper using bigram which divides input sentence as two letters of input sentence. The knowledge of chatbot are stored in the database. The chatbot consists of core and interface that is accessing that core in relational database management systems (RDBMS). The database has been employed as knowledge storage and interpreter has been employed as stored programs of function and procedure sets for pattern-matching requirement.

3.1.1 Drawbacks of Existing System:

1.Complex Interface – Chatbots are often seen to be complicated and require a lot of time to understand user’s requirement. It is also the poor processing which is not able to filter results in time that can annoy people.

2.Inability to Understand – Due to fixed programs, chatbots can be stuck if an unsaved query is presented in front of them. This can lead to customer dissatisfaction and result in loss. It is also the multiple messaging that can be taxing for users and deteriorate the overall experience on the website.

3.Time-Consuming – Chatbots are installed with the motive to speed-up the response and improve customer interaction. However, due to limited data-availability and time required for self-updating, this process appears more time-taking and expensive. Therefore, in place of attending several customers at a time, chatbots appear confused about how to communicate with people.

3.2 Proposed System

In Proposed method,we have implemented bot for college website. Natural Language Processing (NLP) helps provide context and meaning to text-based user inputs so that it can come up with the best response. Implementation of bot and websites for specific college and train data through Rule based algorithm.

3.3 NLP – Natural Language Process

NLP powered chatbots offer great promise for customer savings, and even greater advantages when it comes to costs.

Immediate assistance: Chatbots never sleep and never put you on hold! They are available 24 hours a day, seven days a week for real time interaction.

More efficient service: Chatbots can find the precise answer customers need in any connected knowledge base.

Human-like engagement: Like a human assistant, the chatbot offers a personalized, one-to-one experience, in a conversational style.

Cost and time savings: Chatbots can handle many questions without human intervention, and allows operators to focus on more complex activities.

Natural Language Processing is the driving force behind the following common applications:

1.Language translation applications such as Google Translate

Word Processors such as Microsoft Word and Grammarly that employ NLP to check grammatical accuracy of texts.

2.Interactive Voice Response (IVR) applications used in call centers to respond to certain users' requests.

3.Personal assistant applications such as OK Google, Siri, Cortana, and Alexa.

3.4 FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential. Three key considerations involved in the feasibility analysis are

1.Economic feasibility

2.Technical feasibility

3.Social feasibility

3.4.1 ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

3.4.2 TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand

on the available technical resources. This will lead to high demands being placed on the client. A feasibility study evaluates the project's potential for success.

3.4.3 SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

CHAPTER 4

REQUIREMENT ANALYSIS

4.1HardwareRequirements:

The most common set of requirements defined by any operating system or software application is the physical computer resources, also known as hardware. The minimal hardware requirements are as follows,

System : Pentium IV

Processor speed : 2.4 GHz.

Hard Disk : 1TB

Floppy Drive : 1.44 Mb.

Monitor : 15 VGA Color.

Mouse : Logitech.

Ram : 8Gb.

4.2 Software Requirements:

Software requirements deals with defining resource requirements and prerequisites that needs to be installed on a computer to provide functioning of an application. The minimal software requirements are as follows, Operating system : Windows 10

Coding Language : Python

IDE : Anaconda Navigator

Database : SQL database

4.2.1 CODING LANGUAGES

a) Python

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords

frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

Python is Interpreted − Python is processed at runtime by the interpreter.

Do not need to compile program before executing it. This is similar to PERL and PHP.

Python is Interactive − Can actually sit at a Python prompt and interact with the interpreter directly to write programs.

Python is Object-Oriented − Python supports Object-Oriented style or technique of programming that encapsulates code within objects.

Python is a Beginner's Language − Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

Features

- **Easy-to-learn** − Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
- **Easy-to-read** − Python code is more clearly defined and visible to the eyes.
- **Easy-to-maintain** − Python's source code is fairly easy-to-maintain.
- **A broad standard library** − Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
- **Interactive Mode** − Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
- **Portable** − Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
- **Extendable** − Add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
- **Databases** − Python provides interfaces to all major commercial databases.
- **GUI Programming** − Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
- **Scalable** − Python provides a better structure and support for large programs than shell scripting.

Python Language

Python is an object-oriented programming language created by Guido Rossum in 1989. It is ideally designed for rapid prototyping of complex applications. It has interfaces to many OS system calls and libraries and is extensible to C or C++. Many large companies use the Python programming language include NASA, Google, YouTube, BitTorrent, etc. Python programming is widely used in Artificial Intelligence, Natural Language Generation, Neural Networks and other advanced fields of Computer Science. Python had deep focus on code readability & this class will teach you python from basics.

Python Programming Characteristics

- It provides rich data types and easier to read syntax than any other programming languages
- It is a platform independent scripted language with full access to operating system API's
- Compared to other programming languages, it allows more run-time flexibility
- It includes the basic text manipulation facilities of Perl and Awk
- A module in Python may have one or more classes and free functions
- Libraries in Pythons are cross-platform compatible with Linux, Macintosh, and Windows
- For building large applications, Python can be compiled to byte-code
- Python supports functional and structured programming as well as OOP
- It supports interactive mode that allows interacting Testing and debugging of snippets of code
- In Python, since there is no compilation step, editing, debugging and testing is fast.

4.2.2 Applications of Python Programming

Web Applications

You can create scalable Web Apps using frameworks and CMS (Content Management System) that are built on Python. Some of the popular platforms for creating Web Apps are: Django, Flask, Pyramid, Plone, Django CMS. Sites like Mozilla, Reddit, Instagram and PBS are written in Python.

Scientific and Numeric Computing

There are numerous libraries available in Python for scientific and numeric computing. There are libraries like: SciPy and NumPy that are used in general purpose computing. And, there are specific libraries like: EarthPy for earth science, AstroPy for Astronomy and so on. Also, the language is heavily used in machine learning, data mining and deep learning.

Creating software Prototypes

Python is slow compared to compiled languages like C++ and Java. It might not be a good choice if resources are limited and efficiency is a must. However, Python is a great language for creating prototypes. For example: You can use Pygame (library for creating games) to create your game's prototype first. If you like the prototype, you can use language like C++ to create the actual game.

Good Language to Teach Programming

Python is used by many companies to teach programming to kids and newbies. It is a good language with a lot of features and capabilities. Yet, it's one of the easiest language to learn because of its simple easy-to-use syntax.

About Opencv Package

Python is a general purpose programming language started by Guido van Rossum, which became very popular in short time mainly because of its simplicity and code readability. It enables the programmer to express his ideas in fewer lines of code without reducing any readability.

Compared to other languages like C/C++, Python is slower. But another important feature of Python is that it can be easily extended with C/C++. This feature helps us to write computationally intensive codes in C/C++ and create a Python wrapper for it so that we can use these wrappers as Python modules. This gives us two advantages: first, our code is as fast as original C/C++ code (since it is the actual C++ code working in background) and second, it is very easy to code in Python. This is how OpenCV-Python works, it is a Python wrapper around original C++ implementation.

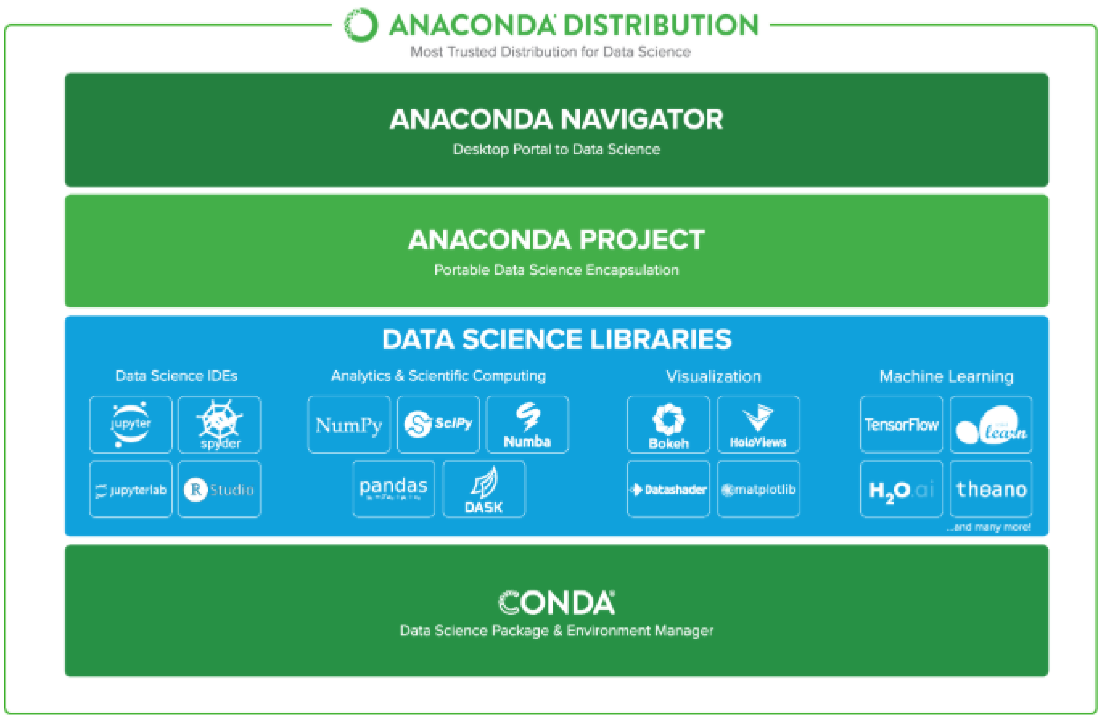
And the support of Numpy makes the task more easier. Numpy is a highly optimized library for numerical operations. It gives a MATLAB-style syntax. All the OpenCV array structures are converted to-and-from Numpy arrays. So whatever operations you can do in Numpy, you can combine it with OpenCV, which increases number of weapons in your arsenal. Besides that, several other libraries like SciPy, Matplotlib which supports Numpy can be used with this.

So OpenCV-Python is an appropriate tool for fast prototyping of computer vision problems.

4.3 FEATURES OF ANACONDA NAVIGATOR

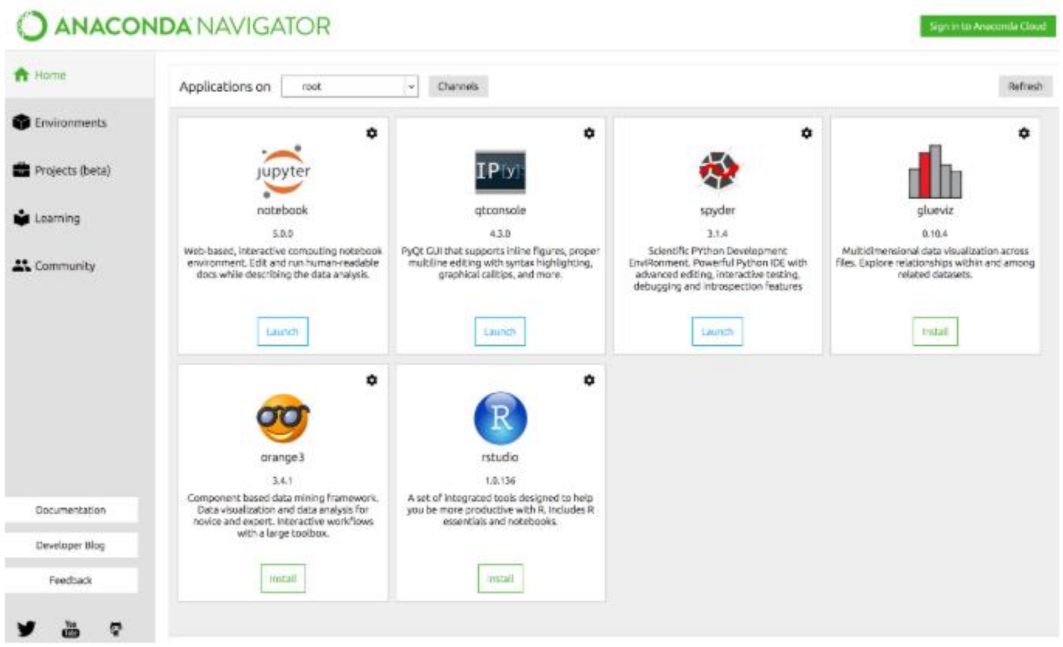
Architecture : Anaconda is a free and open source, easy to install distribution of Python and R programming languages. Anaconda provides a working environment which is used for scientific computing, data science, statistical analysis and machine learning.

The latest distribution of Anaconda is Anaconda 5.3 and is released in October, 2018. It has the conda package, environment manager and a collection of 1000+ open source packages long with free community support.



What is Anaconda Navigator?

Anaconda Navigator is a desktop graphical user interface (GUI) included in the Anaconda distribution. It allows us to launch applications provided in the Anaconda distribution and easily manage conda packages, environments and channels without the use of command-line commands. It is available for Windows, macOS and Linux.



Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.

PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.

Scientific PYthon Development EnviRonment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features

Multidimensional data visualization across files. Explore relationships within and among related datasets.

Component based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows with a large toolbox.

A set of integrated tools designed to help you be more productive with R. Includes R essentials and notebooks.

Anaconda Navigator

Applications Provided in Anaconda Distribution

The Anaconda distribution comes with the following applications along with Anaconda Navigator.

1. JupyterLab

2. Jupyter Notebook

3. Qt Console

4. Spyder

5. Glueviz

6. Orange3

7. RStudio

8. Visual Studio Code

> JupyterLab: This is an extensible working environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.

>Jupyter Notebook: This is a web-based, interactive computing notebook environment. We can edit and run human-readable docs while describing the data analysis.

> Qt Console: It is the PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips and more.

> Spyder: Spyder is a scientific Python Development Environment. It is a powerful Python IDE with advanced editing, interactive testing, debugging and introspection features.

>VS Code: It is a streamlined code editor with support for development operations like debugging, task running and version control.

> Glueviz: This is used for multidimensional data visualization across files. It explores relationships within and among related datasets.

>Orange 3: It is a component-based data mining framework. This can be used for data visualization and data analysis. The workflows in Orange 3 are very interactive and provide a large toolbox.

>Rstudio: It is a set of integrated tools designed to help you be more productive with R. It includes R essentials and notebooks.

NewFeaturesofAnaconda5.3

Compiled with Latest Python release: Anaconda 5.3 is compiled with Python 3.7, taking advantage of Python's speed and feature improvements.

• **Better Reliability:** The reliability of Anaconda has been improved in the latest release by capturing and storing the package metadata for installed packages.

• **Enhanced CPU Peformance:** The Intel Math Kernel Library 2019 for Deep Neural Networks(MKL 2019) has been introduced in Anaconda 5.3 distribution. Users deploying Tensorflow can make use of MKL 2019 for Deep Neural Networks. These Python binary packages are provided to achieve high CPU performance.

• **New packages are added:** There are over 230 packages which has been updated and added in the new release.

• **Work in Progress:** There is a casting bug in Numpy with Python 3.7 but the team is currently working on patching it until Numpy is updated.

CHAPTER 5

SYSTEM DESIGN

5.1 METHODOLOGY:

NLP entails applying algorithms to identify and extract the natural language rules such that the unstructured language data is converted into a form that computers can understand.

When the text has been provided, the computer will utilize algorithms to extract meaning associated with every sentence and collect the essential data from them.

5.2ARCHITECTURE

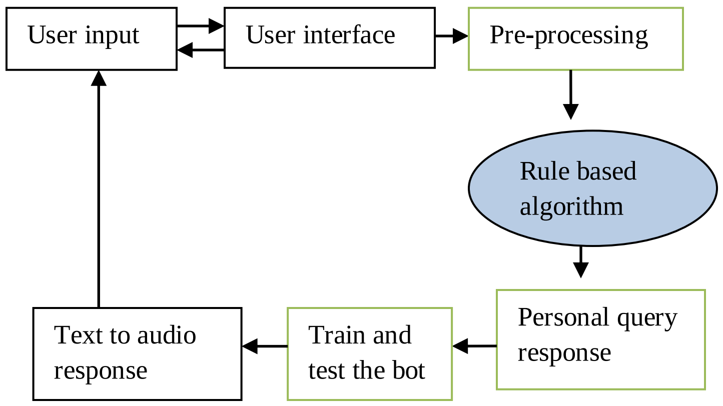


Fig.1. system architecture

Users interact through a device on a messaging platform, his message is processed through NLP.

1. Then the bot can launch an action, answer with realtime information from a database/API, or handover to a human.

2. The more message he receives, the more the bot improves : it's called machine learning. Sometime a human helps the bot, it's called supervised learning.

CHAPTER 6

RESULTS AND PERFORMANCE ANALYSIS

6.1 MODULES

6.1.1Preprocessing:

In this pre-processing, we first load the metadata into this and then this metadata will be attached to the data and replace the converted data with metadata.
Then this data will be moved further and remove the unwanted data in the list and it will divide the data into the train and the test data.

White space tokenization: Non white space sequences are identified by tokens

Simple tokenization: A character class tokenizer,sequences of same character class are tokens.

Learnable tokenization: A maximum entropy tokenizer , detects token boundaries based on probability model.

- ✓ **TOKENIZATION**

Tokenization is essentially splitting a phrase, sentence, paragraph, or an entire text document into smaller units, such as individual words or terms. Each of these smaller units are called tokens. The tokens could be words, numbers or punctuation marks. In tokenization, smaller units are created by locating word boundaries. Tokenization is the first step in text analytics.
The process of breaking down a text paragraph into smaller chunks such as words or sentence is called Tokenization. Token is a single entity that is building blocks for sentence or paragraph.

- ✓ **STEMMING**

Stemming is a process of linguistic normalization, which reduces words to their word root word or chops off the derivational affixes. For example, connection, connected, connecting word reduce to a common word "connect".

- ✓ **LEMMATIZATION**

Lemmatization reduces words to their base word, which is linguistically correct lemmas. It transforms root word with the use of vocabulary and morphological analysis. Lemmatization is usually more sophisticated than stemming. Stemmer works on an individual word without knowledge of the context. For example, The word "better" has "good" as its lemma. This thing will miss by stemming because it requires a dictionary look-up.

6.1.2 Personal Query Response System

Pre-processing is applied to the input text to standardize the input as per the system’s requirement. Based on the keywords used in the text, appropriate context is recognized. Upon receiving personal queries like CGPA, attendance, etc., the authenticity of the user is checked through user-id and password. If the user detail is invalid, an appropriate response is sent. If the user authenticates successfully, the input text is processed to extract keywords. Based on the keywords, information required by the user is understood and the information is provided from the database

6.1.3 Training and testing the dataset

The training data is used to make sure that machine recognises patterns in the data,the cross-validation of the data is used to ensure better accuracy and efficiency of the algorithm used to train machine.Initially, bot has to greet the user who using the website.so,greeting and specified person response is generated using code and their outputs are displayed.There are various approachs for designing a chatbot. In this rule based approach. There is a fixed set of response available and based on a certain rule, a response is selected. For example users says “hello” generate a response ”hi, how are you?”. The main advantage of this approach they often provide perfect response for the user quries. Developing a chatbot is a simple task which is capable of responding to the user queries related to college and will generate answers based on **cosine similarity**. Download the required libraries using NLTK to create our chatbot.

Handling Greetings: We want our chatbot to reply to greetings.For that,we will create a function that handles greetings.We will create two lists with different types of greeting messages.

Response Generation: We need to create a method for general response generation using Cosine similarity.

6.2RULE BASED ALGORITHM

Rules typically take the form of an {IF:THEN} expression, (e.g. {IF 'condition' THEN 'result'}, or as a more specific example, {IF 'red' AND 'octagon' THEN 'stop-sign'}). An individual rule is not in itself a model, since the rule is only applicable when its condition is satisfied. Therefore rule-based machine learning methods typically comprise a set of rules, or knowledge base, that collectively make up the prediction model.

6.3 UML DIAGRAMS

UML is simply another graphical representation of a common semantic model.UML provides a comprehensive notation for the full life cycle of object oriented development. Unified Modeling Language is a general purpose developmental, modeling language in the field of software engineering that is intended to provide a standard way to visualize the design of the system.

UML is a standard language for specifying, visualizing, and documenting of software systems and created by Object Management Group (OMG) in 1997.There are three important type of UML modeling are Structural model, Behavioral model, and Architecture model. To model a system the most important aspect is to capture the dynamic behavior which has some internal or external factors for making the interaction. These internal or external agents are known as actors. It consists of actors, use cases and their relationships. In this fig we represent the Use Case diagram for our project.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

GOALS:

The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.

Be independent of particular programming languages and development process.

3. Provide a formal basis for understanding the modeling language.
4. Encourage the growth of OO tools market.
5. Support higher level development concepts such as collaborations, frameworks, patterns and components.

6.3.1 USECASE DIAGRAM

A use case is a set of scenarios that describing an interaction between a user and a system. A use case diagram displays the relationship among actors and use cases. The two main components a user or another system that will interact with the system modeled. A use case is an external view of the system that represents some action the user might perform in order to complete a task.

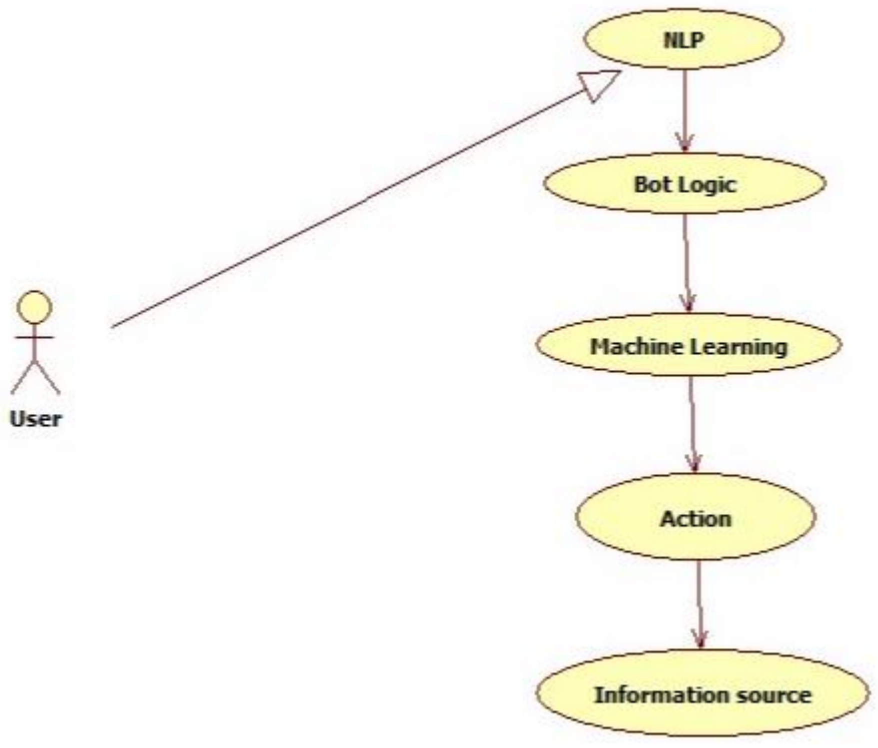


fig.2.usecase diagram

6.2.2 SEQUENCE DIAGRAM

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

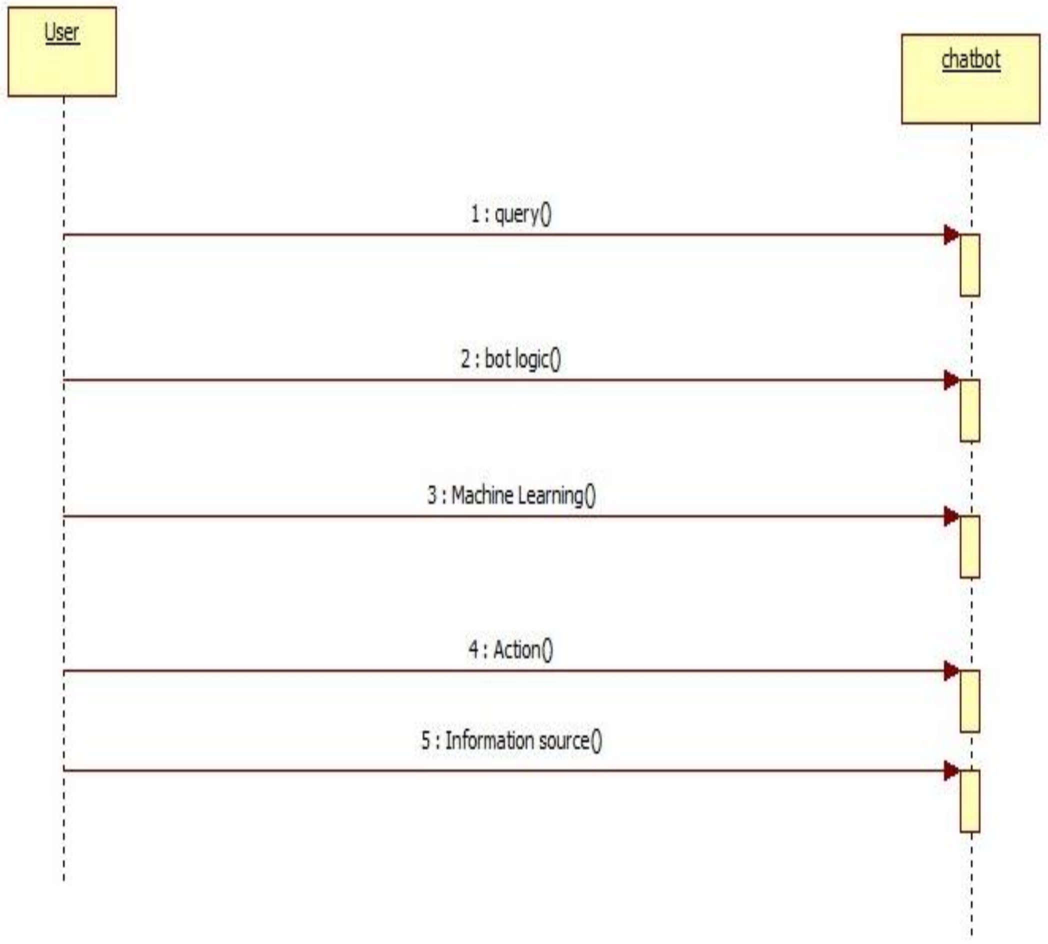


Fig.3. sequence diagram

6.2.3 Activity Diagram

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity

Diagram shows the overall flow of control.

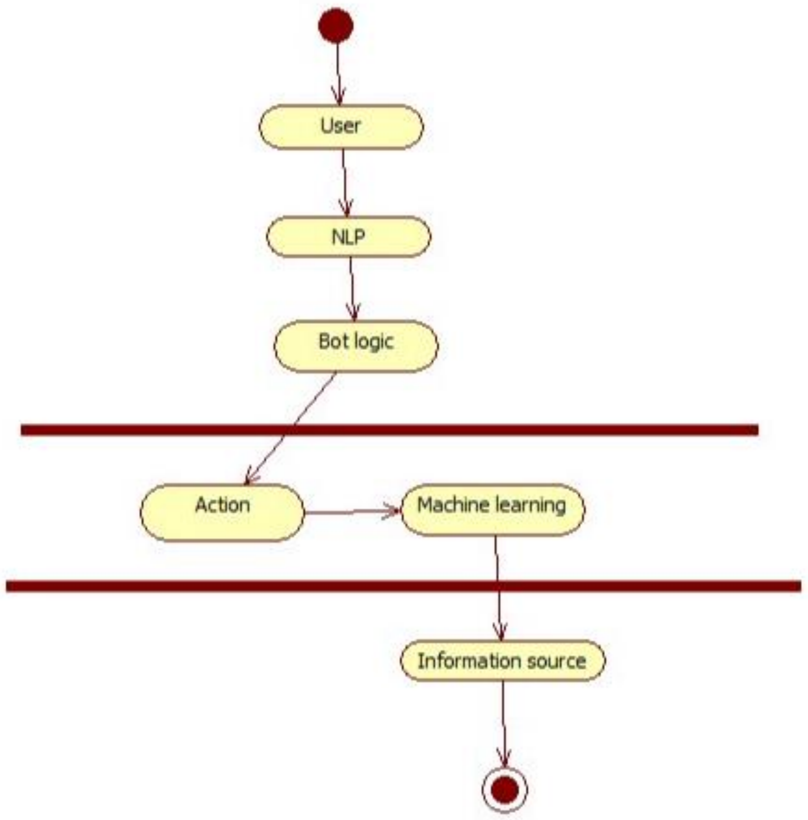


Fig.4.activity diagram

CHAPTER 7

TESTING

7.1 Software Testing

General

In a generalized way, we can say that the system testing is a type of testing in which the main aim is to make sure that system performs efficiently and seamlessly. The process of testing is applied to a program with the main aim to discover an unprecedented error, an error which otherwise could have damaged the future of the software. Test cases which brings up a high possibility of discovering and error is considered successful. This successful test helps to answer the still unknown errors.

7.2 TEST CASE

Testing, as already explained earlier, is the process of discovering all possible weak-points in the finalized software product. Testing helps to counter the working of sub-assemblies, components, assembly and the complete result. The software is taken through different exercises with the main aim of making sure that software meets the business requirement and user-expectations and doesn't fails abruptly. Several types of tests are used today. Each test type addresses a specific testing requirement.

7.3 Testing Techniques

A test plan is a document which describes approach, its scope, its resources and the schedule of aimed testing exercises. It helps to identify almost other test item, the features which are to be tested, its tasks, how will everyone do each task, how much the tester is independent, the environment in which the test is taking place, its technique of design plus the both the end criteria which is used, also rational of choice of theirs, and whatever kind of risk which requires emergency planning. It can be also referred to as the record of the process of test planning. Test plans are usually prepared with signification input from test engineers.

(I) UNIT TESTING

In unit testing, the design of the test cases is involved that helps in the validation of the internal program logic. The validation of all the decision branches and internal code takes place. After the individual unit is completed it takes place. Plus it is taken into account after the individual united is completed before integration. The unit test thus performs the basic level test at its component stage and test the particular business process, system configurations etc. The unit test ensures that the particular unique path of the process gets performed precisely to the documented specifications and contains clearly defined inputs with the results which are expected.

(II) INTEGRATION TESTING

These tests are designed to test the integrated software items to

determine whether if they really execute as a single program or application. The testing is event driven and thus is concerned with the basic outcome of field. The Integration tests demonstrate that the components were individually satisfaction, as already represented by successful unit testing, the components are apt and fine. This type of testing is specially aimed to expose the issues that come-up by the components combination.

(III) FUNCTIONAL TESTING

The functional tests help in providing the systematic representation that functions tested are available and specified by technical requirement, documentation of the system and the user manual.

(IV) SYSTEM TESTING

System testing, as the name suggests, is the type of testing in which ensure that the software system meet the business requirements and aim. Testing of the configuration is taken place here to ensure predictable result and thus analysis of it.System testing is relied on the description of process and its flow, stressing on pre driven process and the points of integration.

(V) WHITE BOX TESTING

The white box testing is the type of testing in which the internal components of the system software is open and can be processed by the tester. It is therefore a complex type of testing process. All the data structure, components etc. are tested by the tester himself to find out a possible bug or error. It is used in situation in which the black box is incapable of finding out a bug. It is a complex type of testing which takes more time to get applied.

(VI) BLACK BOX TESTING

The black box testing is the type of testing in which the internal components of the software is hidden and only the input and output of the system is the key for the tester to find out a bug. It is therefore a simple type of testing. A programmer with basic knowledge can also process this type of testing. It is less time consuming as compared to the white box testing. It is very successful for software which are less complex are straight-forward in nature. It is also less costly than white box testing.

(V) ACCEPTANCE TESTING

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

CHAPTER 8

CONCLUSION AND ENHANCEMENT

8.1 CONCLUSION

Chatbots are extremely valuable for businesses and the value will only increase as time goes by. While the technology to simulate conversation with a computer has been around for decades, bots are adaptable version with a powerful integration of Artificial Intelligence and Natural Language Processing. It creates an assisting guide to all users. It increases the efficiency by maintaining known standard responses. Improved question responsiveness and accuracy. Increased ability to track and monitor queries, highlighting gaps in available information. The impact of NLP by machine will be greater than the impact of microprocessor technology in the last 20 years, because Natural Language is fundamental to almost all business, military & social activities. Therefore, the application of NLP has no end. In future the proposed system will be able interpret the textual description in a much better way.

8.2 FUTURE ENHANCEMENT

In future the Image recognition can be enhanced with much more details about the image captured through the camera. Enhancement to this system can be done by adding the features of currency recognition.

we can make a chatbot which is blend of AIML and LSA. This will enable a client to interact with chatbot in a more natural fashion. We can enhance the 1529 discussion by including and changing patterns and templates for general client queries using AIML and right response are given more often than not utilizing LSA.

APPENDICES

Appendix 1:Sample coding

```
import os

path = r'C:\Users\RECS_02\Desktop\BMC bioinformatics reference removed\2006
text BMC bioinformatics'

for filename in os.listdir(path):

    filen=path+"\\"+filename

    print(filen)

 f=open(filen,'r')

content=f.read()

print(content)

 import glob

 text_files=glob.glob(r"C:\Users\RECS_02\Desktop\BMC bioinformatics reference
removed\2006 text BMC bioinformatics\*.txt")

print(text_files)

for i in text_files[:20]:

    with open(i,"r")as f:

      d=f.read()

 print(d)

 from nltk.tokenize import word_tokenize

nltk.download('punkt')

AI_tokens=word_tokenize(content)

AI_tokens
```

```
len(AI_tokens)

from nltk.probability import FreqDist

fdist=FreqDist()

for word in AI_tokens:

    fdist[word.lower()]+=1

fdist

fdist_top5=fdist.most_common(10)

fdist_top5

from nltk.tokenize import blankline_tokenize

AI_blank=blankline_tokenize(content)

len(AI_blank)

from nltk.util import bigrams,trigrams,ngrams

string="The best and most beautiful things in the world cannot be seen or even
touched, they must be felt with the heart"
quotes_tokens=nltk.word_tokenize(content

quotes_tokens

quotes_bigrams=list(nltk.bigrams(quotes_tokens))

quotes_bigrams

quotes_trigrams=list(nltk.trigrams(quotes_tokens))

quotes_trigrams

quotes_ngrams=list(nltk.ngrams(quotes_tokens,4))

quotes_ngrams

from nltk.stem import PorterStemmer
```

```
pst=PorterStemmer()
pst.stem("restriction")
words_to_stem=["give","giving","given","gave"]
for word in words_to_stem:
print(word+":"+pst.stem(word))
from nltk.stem import wordnet
from nltk.stem import WordNetLemmatizer
nltk.download('wordnet')
word_lem=WordNetLemmatizer()
word_lem.lemmatize("gone")
from nltk.corpus import stopwords
nltk.download('stopwords')
stopwords.words("english")
len(stopwords.words("english"))
fdist_top5
import re
punctuation=re.compile(r'[-.?!,:;()|0-9]')
post_punctuation=[]
for words in AI_tokens:
  word=punctuation.sub("",words)
if len(word)>0:
        post_punctuation.append(word)
post_punctuation
```

```
len(post_punctuation)
 nltk.download('averaged_perceptron_tagger')
 #pos
sent="Restriction enzymes are one of the everyday tools used in molecular
biology."
sent_toke=word_tokenize(sent)
 for token in sent_toke:
    print(nltk.pos_tag([token]))
 from nltk import ne_chunk
nltk.download('maxent_ne_chunker')
 ne_sent="The US President stays in the White House"
nltk.download('words')
 ne_token=word_tokenize(ne_sent)
ne_tags=nltk.pos_tag(ne_token)ne_ner=ne_chunk(ne_tags)
print(ne_ner)
 new="The big cat are the little mouse who was after fresh cheese"
new_toke=nltk.pos_tag(word_tokenize(new))
new_toke


import spacy
nlp=spacy.load('en_core_web_sm')
doc=nlp("hello everyone, I've some good news to give you")
cleaned=[y for y in doc if not y.is_stop and y.pos_!='PUNCT']
```

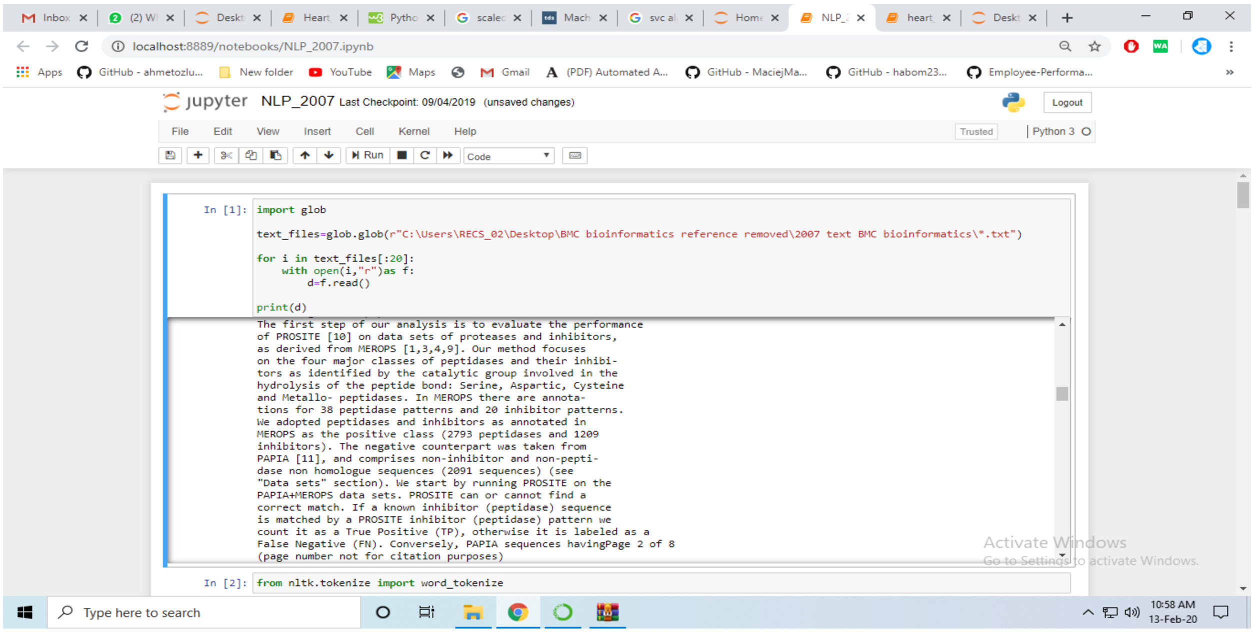
```
raw=[(x.lemma_,x.pos_) for x in cleaned]
print(raw)
to_analyze=('Hello Code & Supply,'
           'My name is Josh and tonight,'
           'We\'re in New York,'
          'by 2020 telecom company Orange')
doc=nlp(to_analyze)
for word in doc.ents:
    print(word.text,word.label_)
spacy.explain("FAC")
import nltk
import warnings
warnings.filterwarnings("ignore")
# nltk.download() # for downloading packages
import numpy as np
import random
import string # to process standard python strings
f=open(r'C:\Users\REtech\Desktop\chatbot.txt','r',errors = 'ignore')
raw=f.read()
raw=raw.lower()# converts to lowercase
#nltk.download('punkt') # first-time use only
#nltk.download('wordnet') # first-time use only
sent_tokens = nltk.sent_tokenize(raw)# converts to list of sentences
```

```
word_tokens = nltk.word_tokenize(raw)# converts to list of words

sent_tokens[:2]

word_tokens[:5]

lemmer = nltk.stem.WordNetLemmatizer()

def LemTokens(tokens):

    return [lemmer.lemmatize(token) for token in tokens]

remove_punct_dict = dict((ord(punct), None) for punct in string.punctuation)

def LemNormalize(text):

    return
LemTokens(nltk.word_tokenize(text.lower().translate(remove_punct_dict)))

GREETING_INPUTS = ("hello", "hi", "greetings", "sup", "what's up","hey","hi
buddy")

GREETING_RESPONSES = ["hi", "hey", "*nods*", "hi there", "hello", "I am
glad! You are

talking to me","hello buddy"]

Welcome_input=("i need project")

Welcome_response=["how can you help you buddy","ddfgdfgdf"]


# Generating response

def response(user_response):

    robo_response=''

    sent_tokens.append(user_response)

    TfidfVec = TfidfVectorizer(tokenizer=LemNormalize, stop_words='english')

    tfidf = TfidfVec.fit_transform(sent_tokens)
    vals = cosine_similarity(tfidf[-1], tfidf)

    idx=vals.argsort()[0][-2]

flat = vals.flatten()

    flat.sort()

    req_tfidf = flat[-2]

if(req_tfidf==0):

        robo_response=robo_response+"I am sorry! I don't understand you"

        return robo_response

    else:

        robo_response = robo_response+sent_tokens[idx]

        return robo_response

 flag=True

print("ROBO: My name is Robo. I will answer your queries about Chatbots. If you
want to exit, type Bye!")


while(flag==True):

user_response = input()

    user_response=user_response.lower()

if(user_response!='bye'):

        if(user_response=='thanks' or user_response=='thank you' ):

            flag=False

            print("ROBO: You are welcome..")

else:
```

```
            if(greeting(user_response)!=None):
                print("ROBO: "+greeting(user_response))
            else:
                print("ROBO: ",end="")
                print(response(user_response))
                sent_tokens.remove(user_response)
else:
        flag=False
        print("ROBO: Bye! take care..")
#imports
from flask import Flask, render_template, request
from chatterbot import ChatBot
from chatterbot.trainers import ChatterBotCorpusTrainer
app = Flask(__name__)
#create chatbot
englishBot = ChatBot("Chatterbot")
trainer = ChatterBotCorpusTrainer(englishBot)
# trainer.train("chatterbot.corpus.english") #train the chatter bot for english
trainer.train("data")
#define app routes
@app.route("/")
def index():
    return render_template("index.html")
```

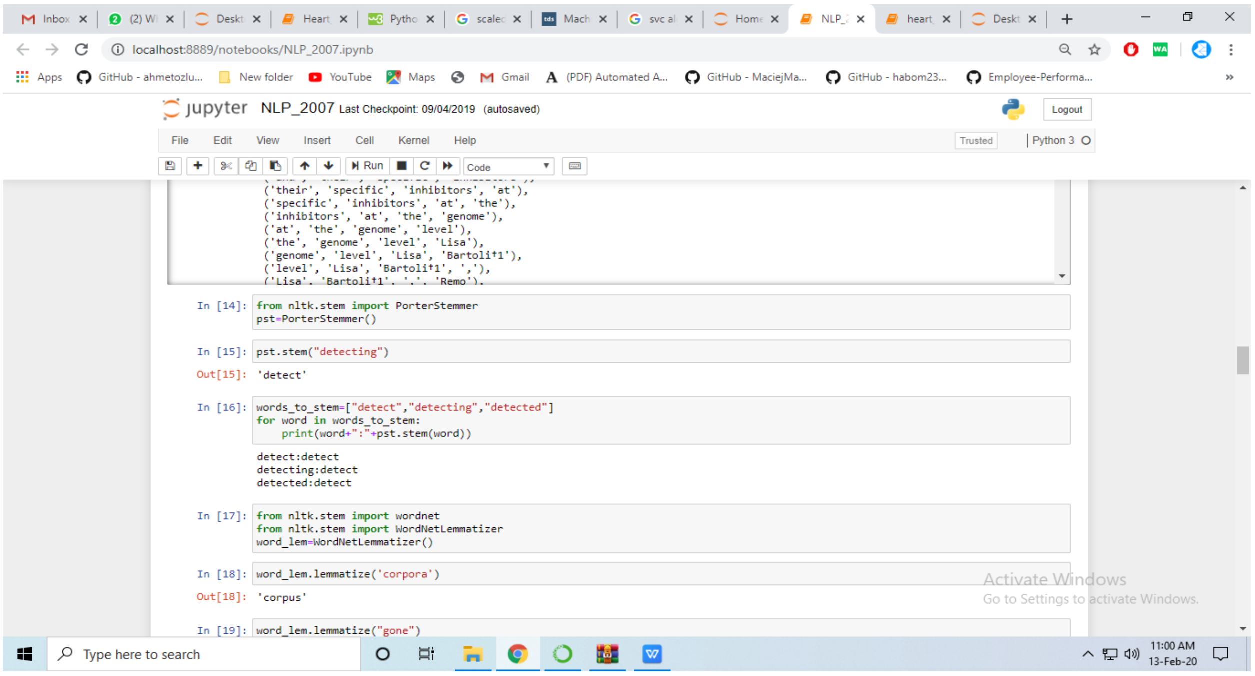
```
@app.route("/get")
#function for the bot response
def get_bot_response():
    userText = request.args.get('msg')
    return str(englishBot.get_response(userText))
if __name__ == "__main__":
   app.run()
```

Appendix 2:Sample output



jupyter NLP_2007 Last Checkpoint: 09/04/2019 (unsaved changes)
localhost:8889/notebooks/NLP_2007.ipynb
File Edit View Insert Cell Kernel Help
Trusted
Python 3
Logout
Run
Code
Type here to search
10:58 AM
13-Feb-20
Activate Windows
Go to Settings to activate Windows.

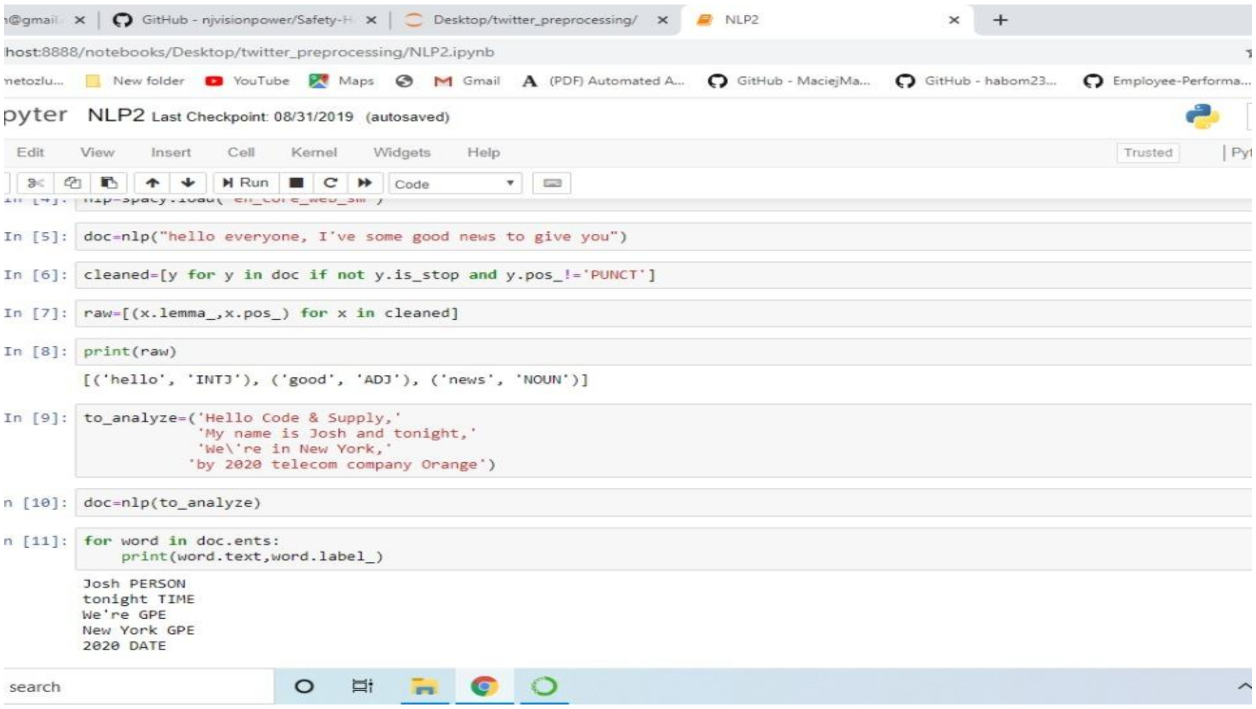
```
In [1]: import glob

        text_files=glob.glob(r"C:\Users\RECS_02\Desktop\BMC bioinformatics reference removed\2007 text BMC bioinformatics\*.txt")

        for i in text_files[:20]:
            with open(i,"r")as f:
                d=f.read()

        print(d)

        The first step of our analysis is to evaluate the performance
        of PROSITE [10] on data sets of proteases and inhibitors,
        as derived from MEROPS [1,3,4,9]. Our method focuses
        on the four major classes of peptidases and their inhibi-
        tors as identified by the catalytic group involved in the
        hydrolysis of the peptide bond: Serine, Aspartic, Cysteine
        and Metallo- peptidases. In MEROPS there are annota-
        tions for 38 peptidase patterns and 20 inhibitor patterns.
        We adopted peptidases and inhibitors as annotated in
        MEROPS as the positive class (2793 peptidases and 1209
        inhibitors). The negative counterpart was taken from
        PAPIA [11], and comprises non-inhibitor and non-pepti-
        dase non homologue sequences (2091 sequences) (see
        "Data sets" section). We start by running PROSITE on the
        PAPIA+MEROPS data sets. PROSITE can or cannot find a
        correct match. If a known inhibitor (peptidase) sequence
        is matched by a PROSITE inhibitor (peptidase) pattern we
        count it as a True Positive (TP), otherwise it is labeled as a
        False Negative (FN). Conversely, PAPIA sequences havingPage 2 of 8
        (page number not for citation purposes)

In [2]: from nltk.tokenize import word_tokenize
```



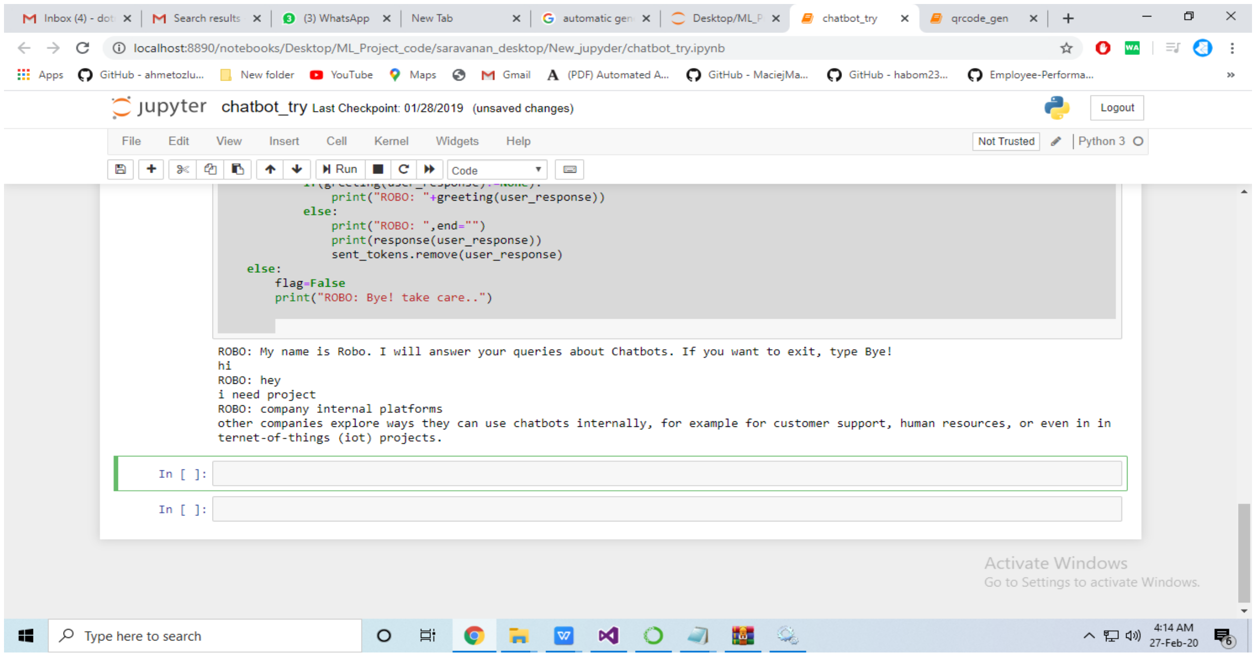
jupyter NLP_2007 Last Checkpoint: 09/04/2019 (autosaved)
localhost:8889/notebooks/NLP_2007.ipynb
File Edit View Insert Cell Kernel Help
Trusted
Python 3
Logout
Run
Code
Type here to search
11:00 AM
13-Feb-20
Activate Windows
Go to Settings to activate Windows.

```
         ('their', 'specific', 'inhibitors', 'at'),
         ('specific', 'inhibitors', 'at', 'the'),
         ('inhibitors', 'at', 'the', 'genome'),
         ('at', 'the', 'genome', 'level'),
         ('the', 'genome', 'level', 'Lisa'),
         ('genome', 'level', 'Lisa', 'Bartoli†1'),
         ('level', 'Lisa', 'Bartoli†1', ','),
         ('Lisa', 'Bartoli†1', ',', 'Remo'),

In [14]: from nltk.stem import PorterStemmer
         pst=PorterStemmer()

In [15]: pst.stem("detecting")

Out[15]: 'detect'

In [16]: words_to_stem=["detect","detecting","detected"]
         for word in words_to_stem:
             print(word+":"+pst.stem(word))

         detect:detect
         detecting:detect
         detected:detect

In [17]: from nltk.stem import wordnet
         from nltk.stem import WordNetLemmatizer
         word_lem=WordNetLemmatizer()

In [18]: word_lem.lemmatize('corpora')

Out[18]: 'corpus'

In [19]: word_lem.lemmatize("gone")
```

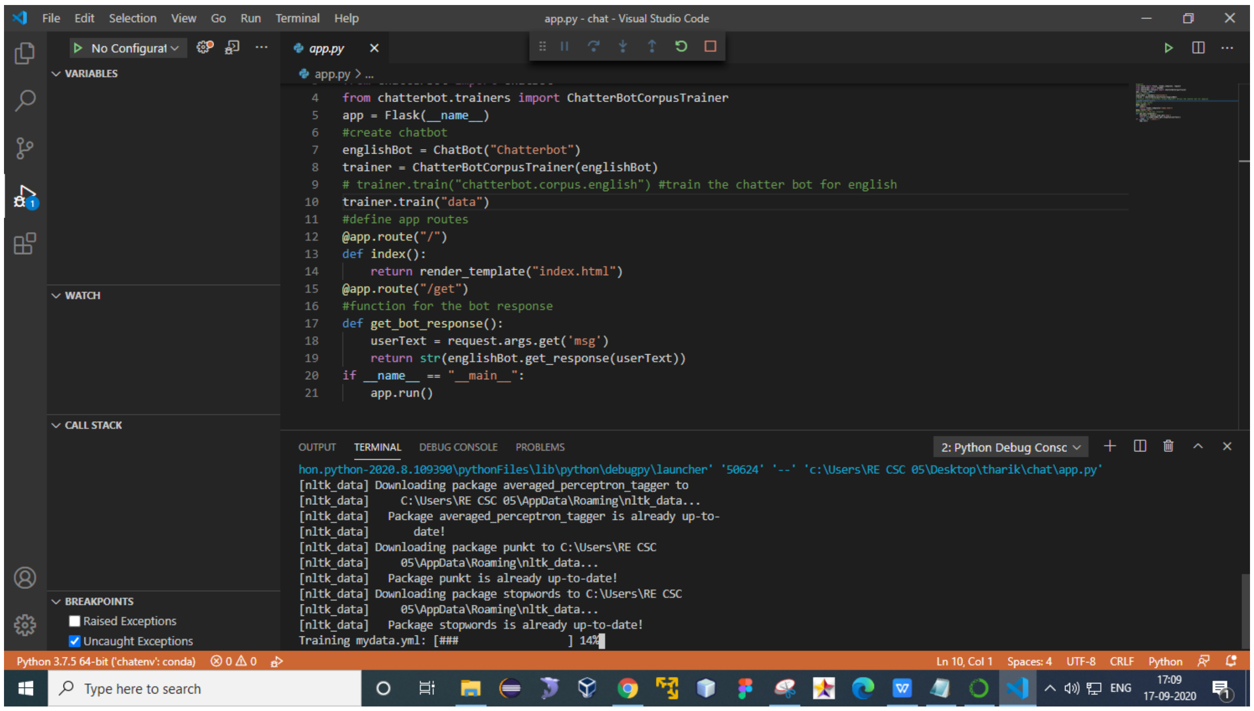
Fig.5. output screenshot



```
In [5]: doc=nlp("hello everyone, I've some good news to give you")

In [6]: cleaned=[y for y in doc if not y.is_stop and y.pos_!='PUNCT']

In [7]: raw=[(x.lemma_,x.pos_) for x in cleaned]

In [8]: print(raw)

        [('hello', 'INTJ'), ('good', 'ADJ'), ('news', 'NOUN')]

In [9]: to_analyze=('Hello Code & Supply,'
                    'My name is Josh and tonight,'
                    'We\'re in New York,'
                   'by 2020 telecom company Orange')

n [10]: doc=nlp(to_analyze)

n [11]: for word in doc.ents:
            print(word.text,word.label_)

        Josh PERSON
        tonight TIME
        We're GPE
        New York GPE
        2020 DATE
```

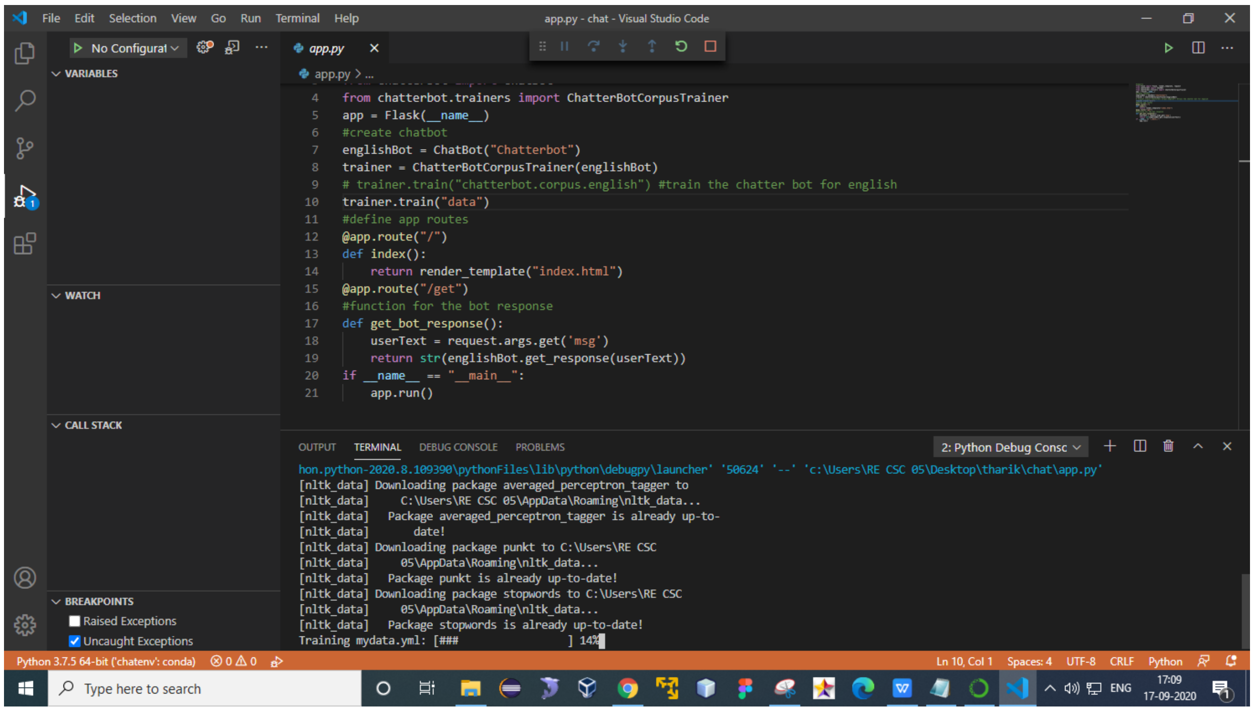


```
                print("ROBO: "+greeting(user_response))
            else:
                print("ROBO: ",end="")
                print(response(user_response))
                sent_tokens.remove(user_response)
    else:
        flag=False
        print("ROBO: Bye! take care..")

ROBO: My name is Robo. I will answer your queries about Chatbots. If you want to exit, type Bye!
hi
ROBO: hey
i need project
ROBO: company internal platforms
other companies explore ways they can use chatbots internally, for example for customer support, human resources, or even in in
ternet-of-things (iot) projects.

In [ ]:

In [ ]:
```



```python
from chatterbot.trainers import ChatterBotCorpusTrainer
app = Flask(__name__)
#create chatbot
englishBot = ChatBot("Chatterbot")
trainer = ChatterBotCorpusTrainer(englishBot)
# trainer.train("chatterbot.corpus.english") #train the chatter bot for english
trainer.train("data")
#define app routes
@app.route("/")
def index():
    return render_template("index.html")
@app.route("/get")
#function for the bot response
def get_bot_response():
    userText = request.args.get('msg')
    return str(englishBot.get_response(userText))
if __name__ == "__main__":
    app.run()
```

```
hon.python-2020.8.109390\pythonFiles\lib\python\debugpy\launcher' '50624' '--' 'c:\Users\RE CSC 05\Desktop\tharik\chat\app.py'
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]     C:\Users\RE CSC 05\AppData\Roaming\nltk_data...
[nltk_data]   Package averaged_perceptron_tagger is already up-to-
[nltk_data]       date!
[nltk_data] Downloading package punkt to C:\Users\RE CSC
[nltk_data]     05\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package stopwords to C:\Users\RE CSC
[nltk_data]     05\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
Training mydata.yml: [###                  ] 14%
```



```python
from chatterbot.trainers import ChatterBotCorpusTrainer
app = Flask(__name__)
#create chatbot
englishBot = ChatBot("Chatterbot")
trainer = ChatterBotCorpusTrainer(englishBot)
# trainer.train("chatterbot.corpus.english") #train the chatter bot for english
trainer.train("data")
#define app routes
@app.route("/")
def index():
    return render_template("index.html")
@app.route("/get")
#function for the bot response
def get_bot_response():
    userText = request.args.get('msg')
    return str(englishBot.get_response(userText))
if __name__ == "__main__":
    app.run()
```

```
hon.python-2020.8.109390\pythonFiles\lib\python\debugpy\launcher' '50624' '--' 'c:\Users\RE CSC 05\Desktop\tharik\chat\app.py'
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]     C:\Users\RE CSC 05\AppData\Roaming\nltk_data...
[nltk_data]   Package averaged_perceptron_tagger is already up-to-
[nltk_data]       date!
[nltk_data] Downloading package punkt to C:\Users\RE CSC
[nltk_data]     05\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package stopwords to C:\Users\RE CSC
[nltk_data]     05\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
Training mydata.yml: [###                  ] 14%
```

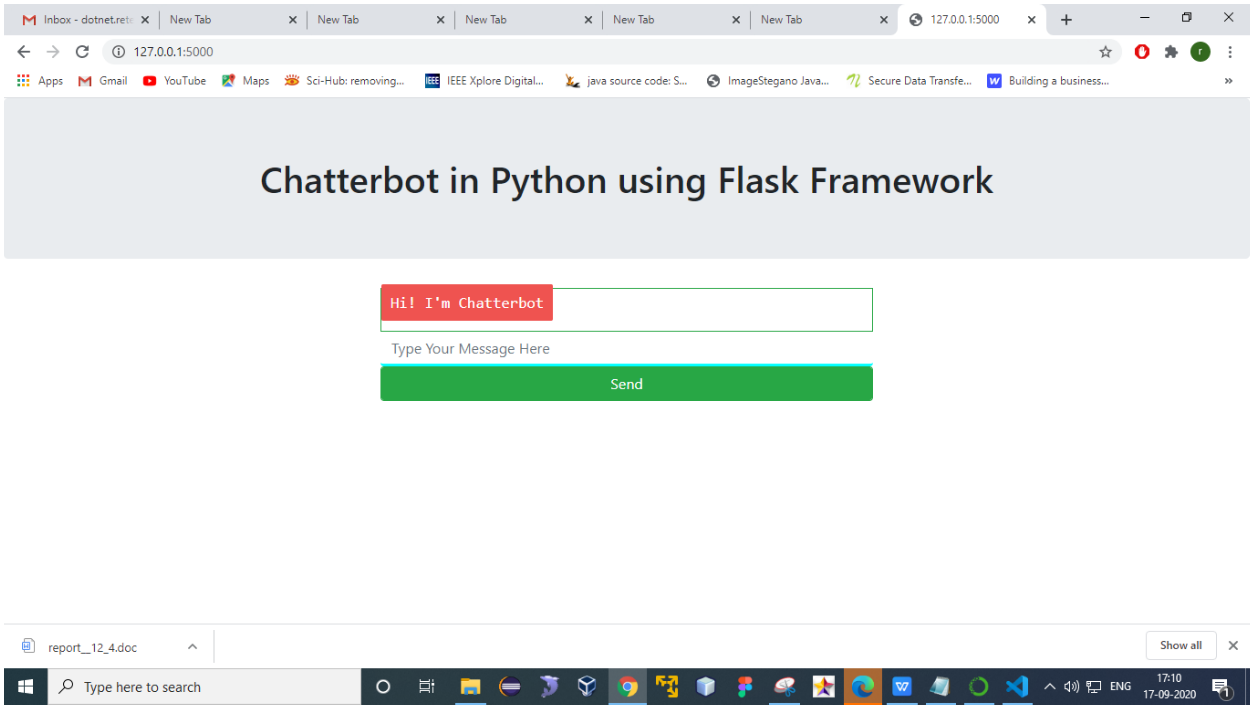


Fig.6.Chatterbot

A3:PUBLICATION

TITLE: Chatbot on college content response using natural language processing.

JOURNAL: International Journal of Advanced development in Science and Technology.

PUBLISHER:IJADST journal vol.2,issue.2,April 2020

ISSN:2582-1059

REFERENCES:

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

2.IntelligentChatbotfor Easy Web-Analytics Insights, Ramya Ravi, 2018, International Conference on Advances in Computing, Communications and Informatics (ICACCI)

3.AI and Web-Based Human-Like Interactive UniversityChatbot(UNIBOT), Neelkumar P. ; Patel Devangi R. ; Parikh Darshan A. ; Patel Ronak R. Patel, 2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA)

4.ChatbotTensorFlow for small Businesses, Rupesh ; Singh Manmath ; Paste Nirmala Shinde; Harshkumar ; Patel Nitin Mishra, 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT)

5. Excitement and Concerns Learning-Based Machine aboutChatbotsand Talkbots: A ; Survey Pablo Rivas Kerstin Holzmayer; Cristian ; Hernandez Charles Grippaldi, 2018 IEEE International Symposium on Technology and Society (ISTAS)

6.Chatbot Learning, Machine using Treatment, Recommendation Rohit Binu ; Mathew ; Sandra Varghese ; Sera Elsa Joy Swanthana Susan Alex, 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI)

7.Designing and Developing aChatbotLearning, Machine Using ; Praveen Kumar Mayank ; Sharma Seema Rawat; Tanupriya Choudhury, 2018

International Conference on System Modeling & Advancement in Research Trends (SMART)

8.Automated Thai-FAQChatbotRNN-LSTM, using Panitan Muangkammuen; Narong Intiruk; Kanda Runapongsa Saikaew, 2018 22nd International Computer Science and Engineering Conference (ICSEC)

9.Intelligent travelchatbotfor predictive recommendation in echo platform, Ashay Argal; Siddharth ; Gupta Ajay Modi; Pratik Pandey; Simon ; Shim Chang Choo, 2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC)

10. Dinus Intelligent Assistance (DINA)Chatbotfor University Admission Services, Heru Agus Santoso; Nurul Anisa Sri Winarsih; Edy Mulyanto; Galuh Wilujeng saraswati; Septian Enggar Sukmana; Supriadi Rustad; Muhammad Syaifur Rohman; Adhitya Nugraha; Fahri Firdausillah, 2018 International Seminar on Application for Technology of Information and Communication