

School of Computer Science and Engineering

Vellore institute of Technology, Chennai

**CSE3013**

**Artificial Intelligence**

CHATBOT FOR VIT WEBSITE

**Submitted by**-

M. Dheeraj Sai - 19BCE1876

K. Manoj Kumar - 19BCE1866

V.D. Bhavesh Kumar - 19BCE1313

**Faculty**: Prof. Padmavathy T V

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| Serial No. | Topic | Page No. |
| 1 | Acknowledgment | 3 |
| 2 | Abstract | 4 |
| 3 | 1. Introduction 2. Why AI is needed for this project 3. PEAS Description, Environment Type, Architecture type 4. Existing System | 5 |
| 4 | Literature Survey | 8 |
| 4 | Proposed Methodology | 10 |
| 4 | Architectural Design | 10 |
| 5 | Modules | 14 |
| 6 | Technologies and Integration | 14 |
| 7 | Implementation Details | 14 |
| 8 | Results and Discussions | 17 |
| 9 | Conclusion and Future Enhancements | 18 |
| 10 | References | 19 |
| 11 | Sample Code | 20 |

**Acknowledgment**

**Primarily,** we would like to thank the almighty for all the blessings he showered over us to complete this project without any flaws.

The success and final outcome of this assignment required a lot of guidance and assistance from many people and we are extremely fortunate to have got this all along with the completion of our project. Whatever we have done is only due to such guidance and assistance by our faculty, Padmavathi T V mam, to whom we are really thankful for giving us an opportunity to do this project.

Last but not the least, we are grateful to all our fellow classmates and our friends for the suggestions and support given to us throughout the completion of our project.

**Abstract**

Question answering (QA) is intended to answer questions defined in natural language. The question answering system provides an automated approach to getting a solution for natural language queries. Many QA surveys classify question-answering systems based on a variety of criteria, including: A more comprehensive overview of the QA scheme is essential to a complete understanding of the QA scheme, how it has grown to meet current QA requirements, and the need to expand to meet future expectations. In this project we will do a quick survey of the general QA framework in terms of question analysis, passage search, answer extraction, and some important questions related to the QA system.

To utilise software programmes, we can use command line, graphical user interface (GUI), menu driven, form-based, natural language, and other user interfaces. Although the most common user interfaces are GUI and web-based, there are times when a different user interface is required. This is where a conversational user interface chat bot based on Question Answering System comes in handy. They usually offer a stateful service, which implies that each session's data is preserved.

It's common to be unsure where to go on collage website for information. The answer is an inquiry chat bot, which is a quick, simple, and informative widget that improves collage website's user experience while also giving important information to the user. Chat bots are sophisticated algorithms that communicate with humans using artificial intelligence (AI) and natural language processing (NLP) techniques.

**Introduction**

Question Answering (QA) is intended to automatically provide a solution for natural language queries. Question answering systems aim to get the expected answer to a question, rather than ranking the documents as in most information retrieval systems. The idea of ​​a question answering system shows remarkable progress in information retrieval technology, especially in the ability to naturally access knowledge resources by simply querying and retrieving the correct answer.

Our Chat Bot is a computer program that can talk to humans in natural language, the way we interact with each other. It can replace a human for many queries response tasks. A chatbot is an agent that interacts with the user using natural language. One of the main goals of our chatbot is to look like a human and make it difficult for the recipient of the conversation to understand how it actually works and provide related useful information for the queries that the recipient has posted.

Chatbot enable customers to have a better experience by simplifying complex services and interactions. These chatbots can trick users into believing they are "talking" to a human, but they are limited in their ability to build their knowledge base in real time. For this limitation Chatbots makes use of machine learning to reach artificial intelligence helping them to understand the user query and provide an appropriate response. The chatbots are developed using the Artificial Intelligence Markup Language for communicating or interacting with the user. This consist a software which will be made up using Artificial Intelligence and will help user to chat with machine.

**Why AI is needed for this Project**

The main feature of a chatbot is to have a human like conversation with the user. A chatbot should be very intelligent in handling the user queries and respond with an accurate related answer. A chatbot using AI is more advanced and can understand open-ended queries. AI chatbots use natural language processing and machine learning algorithms to become smarter over time. They are more akin to an actual live representative that can grow and gain more skills. AI is needed to retrieve collage website’s information faster, save time for the user, and minimize website traffic.

AI chatbots can improve their functionality and become smarter as time progresses. They can learn new features and adapt as required. Intelligent chatbots become more intelligent over time using NLP and machine learning algorithms. Well programmed intelligent chatbots can gauge a website visitor’s sentiment and temperament to respond fluidly and dynamically.

**PEAS Description**

Performance Measure – Accuracy of the chatbot, Relevance of the retrieved information to query, Retrieval Time of the information.

Environment – VIT Website, Website End-User

Actuators – Information or relevant information retrieved based on the query posted by the user into the chatbot.

Sensors – Keyboard, Mouse to post Query.

**Environment Types**

1. Fully Observable: As the agent sensor is capable to sense or access the complete state of an agent at each point in time.

2. Deterministic: As we can completely determine the next state of the environment from current state.

3. Sequential: As agent requires memory of past actions to determine the next best actions.​

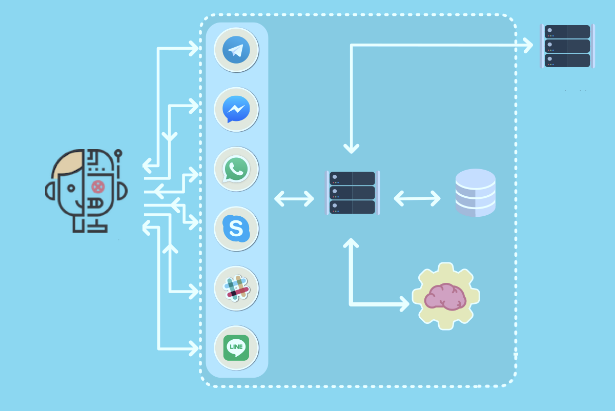
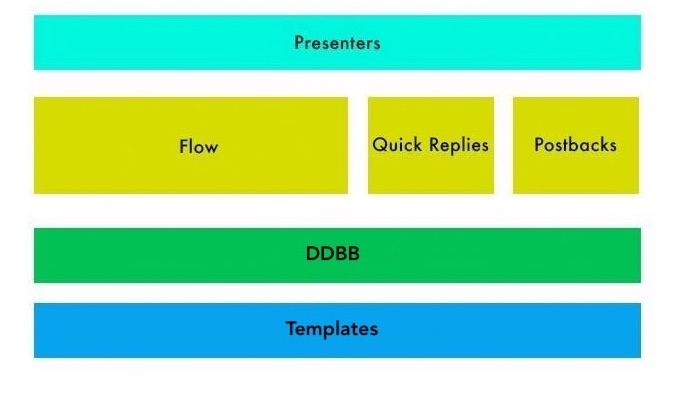
4. Semi Dynamic: As environment itself does not change with the passage of time but the agent's performance score does.​

5. Continuous: As the action performed cannot be numbered.

6. Single Agent: As only one agent is involved.

**Architectural Type**

AI / ML driven architecture

**** ****

**Existing System**

Most of the chatbots used today are used for one specific application : Information Retrieval. This class of bots is designed to provide human-like answers without human intervention. Here, the system tries to understand what question you are trying to ask, or more realistically which question from it’s bank is closest to the question you are trying to ask, and answers it accordingly (Pre stored and/or trained). Of course internet variants exist where you search online or query a database for results, or maybe even run diagnostics, analytics or some other functionality. They are gaining importance by the day and have a great lot of development put into them.

There are mostly 3 types of chatbots:

1. Rule Based
2. Deep Learning Based
3. The Hybrid model, which is a combination of the above two.

**1. Rule Based Chatbot**

Rule-based chatbots are also referred to as decision-tree bots. As the name suggests, they use a series of defined rules. These rules are the basis for the types of problems the chatbot is familiar with and can deliver solutions for. Rule-based chatbots can use very simple or complicated rules.

Advantages of Rule Based Chatbot

* + They are generally faster to train (less expensive)
  + They integrate easily with legacy systems
  + They streamline the handover to a human agent
  + They are highly accountable and secure
  + They can include interactive elements and media
  + They are not restricted to text interactions

**2. Deep Learning Based Chatbot**

AI chatbots that use machine learning and deep learning understand the context and intent of a question before formulating a response. These chatbots generate their own answers to more complicated questions using natural-language responses. The more you use and train these bots, the more they learn and the better they operate with the user.

Advantages of Deep Learning Chatbots –

* + They learn from information gathered
  + They continuously improve as more data comes in
  + They understand patterns of behavior
  + They have a broader range of decision-making skills
  + They can understand many languages

**3. Hybrid Model based Chatbot.**

They are a combination of rule based and AI chatbots. Both rule based and AI chatbots are extremes in the chatbot spectrum. There will constantly be a need for rule based chatbots to be smarter and AI chatbots to be simpler. Hybrid chatbots meet that middle ground. Hybrid chatbots have some rule-based tasks, and they can understand intent and context. This makes them a balanced tool for businesses to interact with customers.

**Literature Survey**

1. AI BASED CHATBOT, Prof. Nikita Hatwar, Ashwini Patil, Diksha Gondane International Journal of Emerging Trends in Engineering and Basic Sciences (IJEEBS) ISSN (Online) 2349-6967 Volume 3, Issue 2 (March-April 2016) - This paper demonstrates the creation of an artificially intelligent chatter bot with whom humans may communicate by speaking to it and receiving a response from the bot via its voice synthesiser. The goal of this study is to demonstrate how a chatter bot may be utilised in a variety of disciplines, including education, healthcare, and route help. It is a statistical model and chatter bot that uses Microsoft voice synthesiser for speech recognition and natural language processing and is trained using AIML (Artificial Intelligent Markup Language) structure.
2. Adoption of AI-based chatbots for Hospitality and Tourism, Rajasshrie Pillai, Brijesh Sivathanu, Pune, India (2020). This research provides critical insights for practitioners and managers by providing the factors of AIN of chatbots in tourism. This study highlights some of the manager’s viewpoints, which can be considered to comprehend the adoption of chatbots in the tourism industry. Marketers and designers of tourism chatbots need to ensure that chatbots are easy and simple to use as well as useful in the tour planning and travel management.
3. AI-based chatbots in customer service and their effects on user compliance, Martin Adam1 & Michael Wessel2 & Alexander Benlian1, (2020) In this study, they conducted an online experiment to show that both verbal anthropomorphic design cues and the foot-in-the-door technique increase user compliance with a chatbot’s request for service feedback. their study is thus an initial step towards better understanding how AI-based CAs may improve user compliance by leveraging the effects of anthropomorphism and the need to stay consistent in the context of electronic markets and customer service.
4. Parsing and Question Classification for Question Answering, Ulf Hermjakob Information Sciences Institute University of Southern California, (2017) This paper describes machine learning based parsing and question classification for question answering. We demonstrate that for this type of application, parse trees have to be semantically richer and structurally more oriented towards semantics than what most treebanks offer. They empirically show how question parsing dramatically improves when augmenting a semantically enriched Penn treebank training corpus with an additional question treebank.
5. Named Entity Recognition in Question Answering of Speech Data, Diego Molla´ Menno van Zaanen Steve Cassidy, (2007), In this paper they present our contribution to QAst, which is centred on a study of Named Entity (NE) recognition on speech transcripts, and how it impacts on the accuracy of the final question answering system. they have ported AFNER, the NE recogniser of the Ans they Finder question answering project, to the set of ans they types expected in the QA track. AFNER uses a combination of regular expressions, lists of names (gazetteers) and machine learning to find Ne in the data.
6. ApplyingWord Sense Disambiguation to Question Answering System for E-Learning, Jason C. Hung, Che-Yu Yang, (2005), In this paper, they introduce a semantic-based automated question ans they ring system that can act like a virtual teacher to respond to student questions online. With the system, not only the instructor can be relieved from the load of ans they ring lots of questions, but also the student can mostly get ans they promptly without waiting for the instructor to get online and provide an ans they. This would be a big help for both the instructor and the student in e-learning environment.

**Need for a New System -**

The need for a university inquiry system is due to a variety of reasons including the slow nature of the university website, a stranger wouldn't know where to look for specific information that would be difficult for him or her who is outside the university to go through the university information. A smart solution for all disadvantages shows the need for a question answering system.

**Proposed System**

We propose a question answering system based chatbot which can be embedded into the college website which helps to solve the drawbacks of a traditional collage website where users or students find it difficult to find the information which they are looking. The university question answering system based Chatbot will provide an answer by summarizing the question and providing the relevant answer. It also provides selective information about what the user wants.

A university chatbot will provide all the answers related to areas like admission, test box, bulletin board, attendance, placement box and others. We have taken a lot of pre-defined questions relating to all the possible queries that a user would like to know or ask a chatbot in a collage website or university perspective domain. And in the pre-defined query dataset we have divided all the queries or questions into a certain class where they would fit and for all the available question sentences, we have divided it into unique lemmatized words and when a query is posted the ML algorithm will take the query and find all the unique lemmatized words and match it with the available query dataset and will classify it into any one of the available classes and respond with a suitable more precise answer.

**Architectural Design**

An architecture of Chatbot requires a candidate response generator and response selector to give the response to the user’s queries through text, images, URL’s, documents, etc.

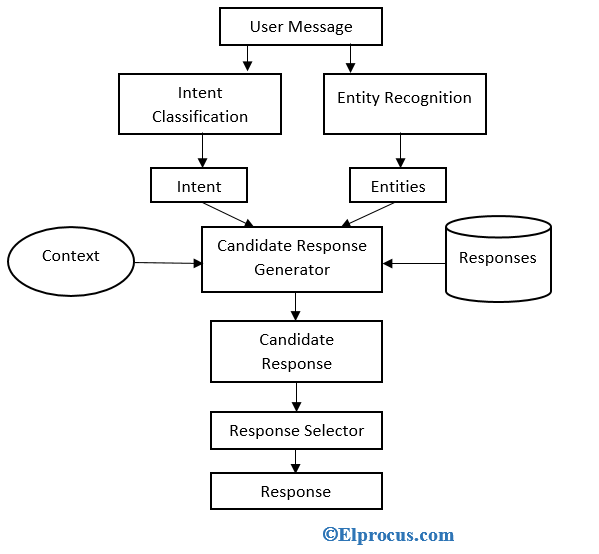


Fig.1. System Architecture for the proposed System

In the above figure, user messages are given to an intent classification and entity recognition.

**Intent:** An intent in the above figure is defined as a user’s intention, example the intent of the word “Good Bye” is to end the conversation similarly, the intent of the word “where is the collage located” the intent would be to find collage location in Google Maps.

**Entity:** The entity extractor is responsible for identifying keywords from the user’s query that helps determine what the user is looking for.

**Candidate Response Generator:** The candidate response generator in the Chatbot do the calculations using different algorithms to process the user request. Then the result of these calculations is the candidate’s response.

**Response Selector:** The response selector in the Chatbot used to select the word or text according to the user queries to give a response to the users which should work better.

**Working of the Model -**

So, the flow of the chatbot is as follows:

1. First we have a file called 'intents.json', in this file we have a set of predefined questions and answers. All questions and answers can be separated by 'tags'. 'tags' here means a specific topic or category of a question. For ex, tag=Canteen means the question asked by the user falls into canteen category.
2. We have a neural network which will train on the questions, obviously we need to convert text into bag-of-words representation in order to do that. Here, we will learn the association between the question and the tag. Which means our feature variable is Bag of words representation of the questions and our target variable is the tag associated with those questions. Yes, it is a classification task.
3. Once we find the tag, we just need to return an answer for that tag entry through the json file.

A unique pattern must be available in the database to provide a suitable response for each kind of question. A hierarchy is created with lots of combinations of patterns. Algorithms are used to reduce the number of classifiers and create a more manageable structure. This is called Reductionist approach.

For example, let us see at the set of sentences that belong to a particular class. With new input sentences, each word is counted for its occurrence and is accounted for its commonality. Then, each class is assigned a score. The highest scored class is the most likely to be associated with the input sentence.

**Example of Sample Training Set**:

Class: Greetings

*“How are you doing?”“Good morning”*

*“Hi, there!”*

**Sample Input Sentence Classification**:

Input: “Hello, good morning.”

*Term: “Hello” (no matches)*

*Term: “Good” (class: Greetings)*

*Term: “morning” (class: Greetings)*

*Classification: Greetings (score=2)*

With the help of an equation, word matches are found for the given sample sentences for each class. The classification score identifies the class with the highest term matches, but it also has some limitations. The score signifies which intent is most likely to the sentence but does not guarantee it is the perfect match. The highest score only provides the relativity base.

**Artificial Neural Networks**

Neural Networks are a way of calculating the output from the input using weighted connections, which are computed from repeated iterations while training the data. Each step through the training data amends the weights resulting in the output with accuracy.

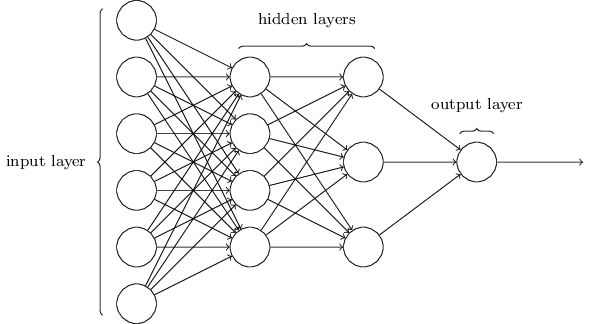


Fig.2 Neural Network Working - Source- marutitech.com

As discussed earlier here, each sentence is broken down into individual words, and each word is then used as input for the neural networks. The weighted connections are then calculated by different iterations through the training data thousands of times, each time improving the weights to make it accurate.

**NLU (NATURAL LANGUAGE UNDERSTANDING)**

NLU helps the chatbot understand the query by breaking it down. We have used two specific concepts in building our college chatbot:

1. **Entities**: An entity represents keywords from the user’s query picked up by the chatbot to understand what the user wants. It is a concept in your chatbot. E.g., ‘Where is VIT’s Location?’ has the word ‘Location’ as an entity.
2. **Intents**: It helps identify the action the chatbot needs to perform on the user’s input. For instance, the intent of “Where is VIT” and “What is VIT Location? and “Where is VIT located” is the same. All of these user’s texts trigger a single command giving users options for t-shirts.

**NLP (NATURAL LANGUAGE PROCESSING)**

Natural Language Processing (NLP) chatbot takes some steps to convert the customer’s text or speech into structured data to select the related answer. Some of the Natural Language Processing steps which we have used are:

* **Tokenization:** The NLP divides a string of words into pieces or tokens. These tokens are linguistically symbolic or are differently helpful for the application.
* **Named Entity Recognition:** The chatbot program model looks for categories of words, like the name of the product, the user’s name or address, whichever data is required.

Like most applications, the chatbot is also connected to the database. The knowledge base or the database of information is used to feed the chatbot with the information required to give a suitable response to the user.

The information about whether or not your chatbot could match the users’ questions is captured in the data store. NLP helps translate human language into a combination of patterns and text that can be mapped in real-time to find appropriate responses.

**Modules**

Chatbot for VIT Website is divided into the following modules:

1. Input query pre-processing
2. Database Creation
3. Response Generator
4. Neural Networks
5. Training of the Model

**Technologies and Integration**

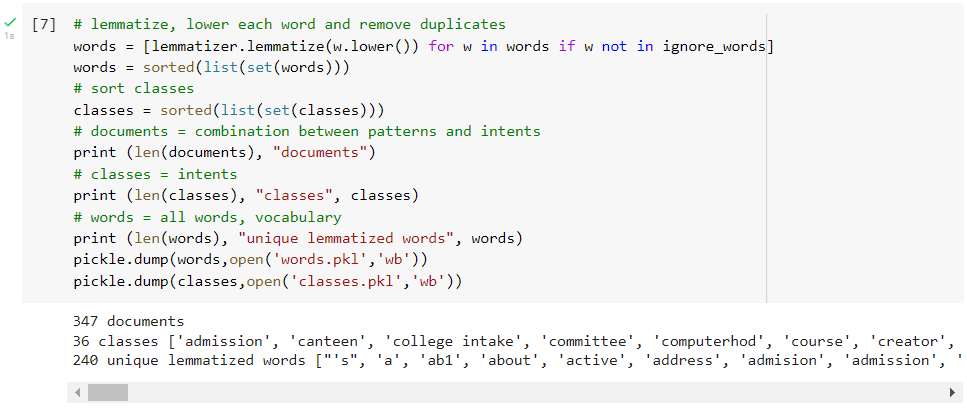
The tools and programming languages used in this project are as follows:

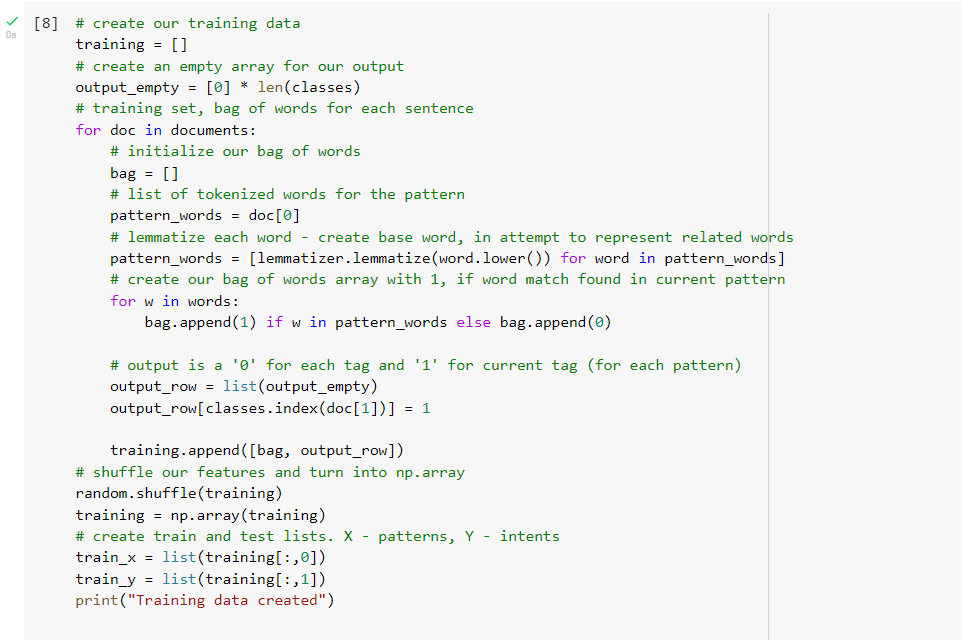
1. Python
2. Python notebook
3. NLTK library (NLP)
4. Json (Data Base)
5. Pickle library
6. Numpy
7. Keras (Neural Network)
8. Google Colab (to run the ipynb file and connect the required files and Databse)

**Implementation Details**

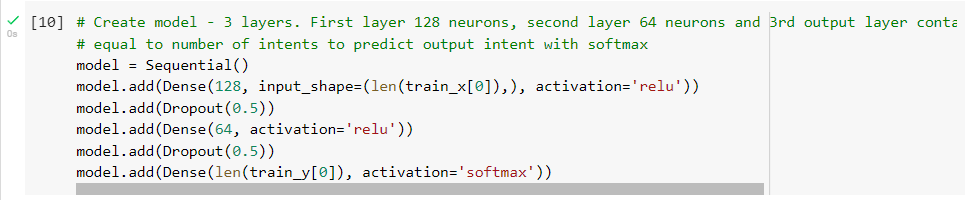
Pre-processing:



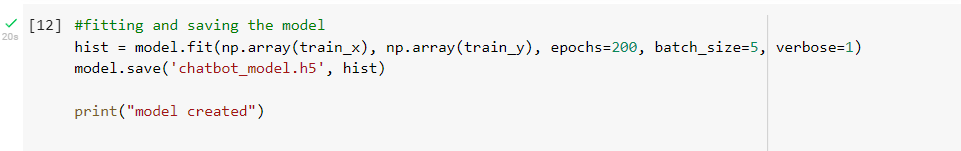


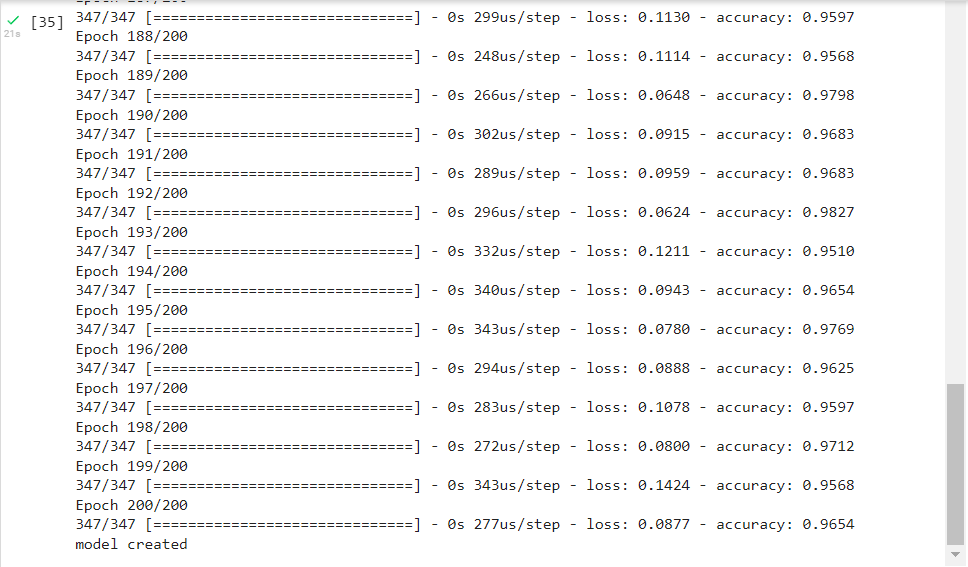


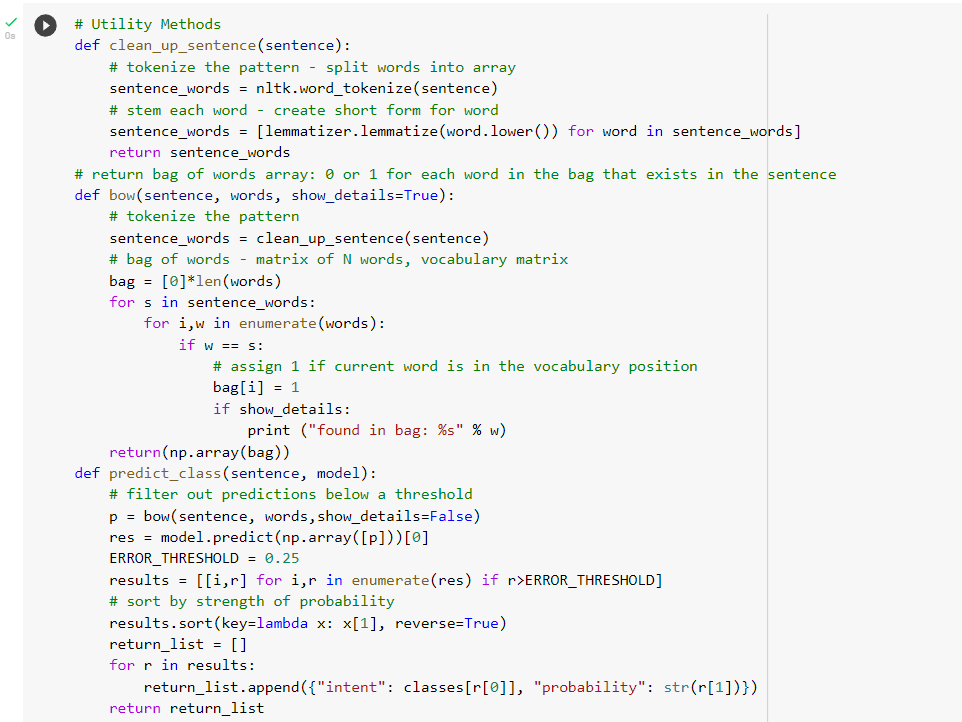
Neural Networks



Training of Data

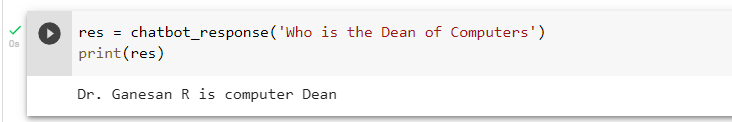
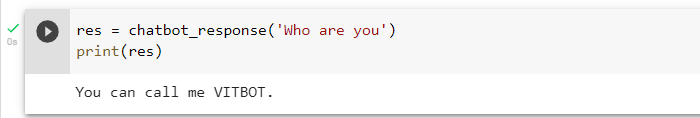
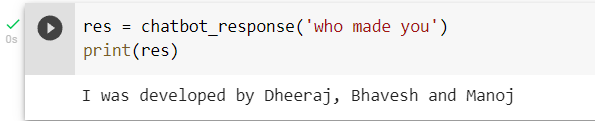




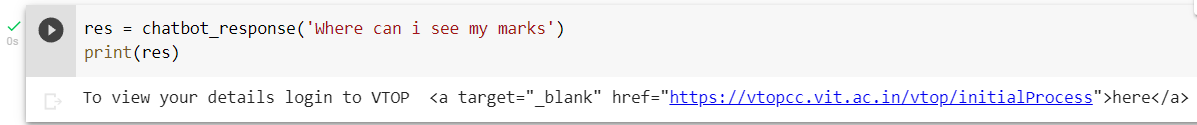


**Results and Discussions**

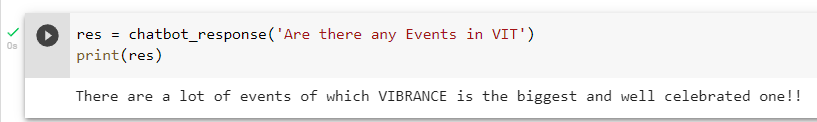
Sample Queries and Answers given by the Chatbot





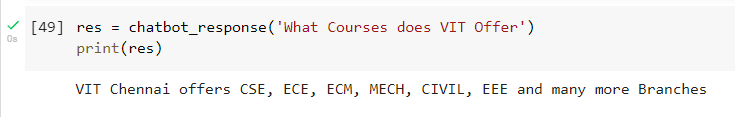


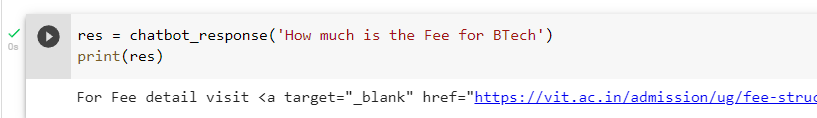




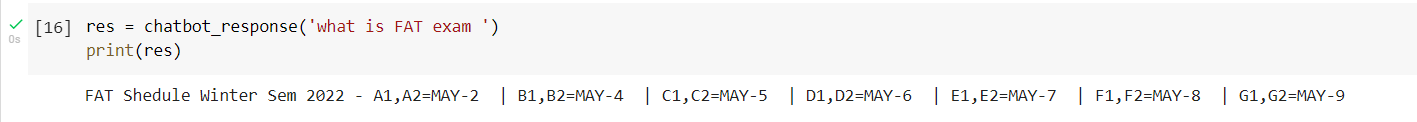


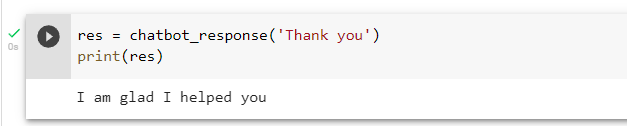












**Conclusion and Future Enhancements**

Artificial Intelligence (AI) is the world's fastest-growing technology. Using Artificial Intelligence and Knowledgeable Databases. We have the ability to alter pattern matching and help virtually.

This method is building a chatbot for a university in order to reduce website traffic. It is made up of a mix of artificial intelligence, knowledge databases, and virtual assistance. We can build a chat bot that transforms between human and machine speech and responds to the user's question. For the future works we can try to integrate the chatbot onto the VIT website and also integrate voice recognition to ask question using speech

**References**

1. AI BASED CHATBOT Prof.Nikita Hatwar 1, Ashwini Patil , Diksha Gondane 3 123 (Information Technology, Priyadarshini College of Engineering,Nagpur/ RTMNU,India)International Journal of Emerging Trends in Engineering and Basic Sciences (IJEEBS) ISSN (Online) 2349-6967 Volume 3, Issue 2 (March-April 2016)
2. Y. Wu, G. Wang, W. Li, and Z. Li, “Automatic Chatbot Knowledge Acquisition from Online Forum via Rough Set and Ensemble Learning,” Proc. IEEE of 2008 IFIP International Conference on Network and Parallel Computing, 2008, pp. 242-246, doi:10.1109/NPC.2008.24.
3. J. Bang, H. Noh, Y. Kim and G. G. Lee, "Example-based chatoriented dialogue system with personalized long-term memory,"2015 International Conference on Big Data and Smart Computing (BIGCOMP), Jeju, 2015
4. Y. Chen, W. Wang and Z. Liu, "Keyword-based search and exploration on databases," 2011 IEEE 27th International Conference on Data Engineering, Hannover, 2011
5. Sanda Harabagiu, Marius Pasca and Steven Maiorano 2000. Experiments with Open-Domain Textual Question Answering In Proceedings of COLING2000, Saarbrucken.
6. Srihari, C. Niu, and W. Li. 2000. A Hybrid Approach for Named Entity and Sub-Type Tagging. In Proceedings of the conference on Applied Natural Language Processing (ANLP 2000), Seattle.
7. Ed Hovy, L. Gerber, U. Hermjakob, C.-Y. Lin, D. Ravichandran 2001. Towards Semantics-Based Answer Pinpointing In Proceedings of the HLT 2001 Conference, San Diego
8. Wu M., Zheng X., Duan M., Liu T. and Strzalkowski T., “Question Answering by Pattern Matching, WebProofing, Semantic Form Proofing”. NIST Special Publication: The Twelfth Text REtrieval Conference (TREC),p. 500-255. (2003)
9. Laurent D., Séguéla P. and NègreCross S., “Lingual Question Answering using QRISTAL for CLEF 2006“. Lecture Notes in Computer Science, Vol. 4730, 2007, pp. 339-350.

**Sample Code**

!pip install nltk

import nltk

nltk.download('punkt')

import nltk

from nltk.stem import WordNetLemmatizer

lemmatizer = WordNetLemmatizer()

import json

import pickle

import warnings

warnings.filterwarnings('ignore')

%tensorflow\_version 1.x

import numpy as np

from keras.models import Sequential

from keras.layers import Dense, Activation, Dropout

from keras.optimizers import SGD

import random

words=[]

classes = []

documents = []

ignore\_words = ['?', '!']

data\_file = open('/content/intents.json').read() # you can find the intents file here:https://github.com/Sanket758/AI-Chatbot-with-Tensorflow

intents = json.loads(data\_file)

for intent in intents['intents']:

    for pattern in intent['patterns']:

        #tokenize each word

        w = nltk.word\_tokenize(pattern)

        words.extend(w)

        #add documents in the corpus

        documents.append((w, intent['tag']))

        # add to our classes list

        if intent['tag'] not in classes:

            classes.append(intent['tag'])

nltk.download('wordnet')

# lemmatize, lower each word and remove duplicates

words = [lemmatizer.lemmatize(w.lower()) for w in words if w not in ignore\_words]

words = sorted(list(set(words)))

# sort classes

classes = sorted(list(set(classes)))

# documents = combination between patterns and intents

print (len(documents), "documents")

# classes = intents

print (len(classes), "classes", classes)

# words = all words, vocabulary

print (len(words), "unique lemmatized words", words)

pickle.dump(words,open('words.pkl','wb'))

pickle.dump(classes,open('classes.pkl','wb'))

# create our training data

training = []

# create an empty array for our output

output\_empty = [0] \* len(classes)

# training set, bag of words for each sentence

for doc in documents:

    # initialize our bag of words

    bag = []

    # list of tokenized words for the pattern

    pattern\_words = doc[0]

    # lemmatize each word - create base word, in attempt to represent related words

    pattern\_words = [lemmatizer.lemmatize(word.lower()) for word in pattern\_words]

    # create our bag of words array with 1, if word match found in current pattern

    for w in words:

        bag.append(1) if w in pattern\_words else bag.append(0)

    # output is a '0' for each tag and '1' for current tag (for each pattern)

    output\_row = list(output\_empty)

    output\_row[classes.index(doc[1])] = 1

    training.append([bag, output\_row])

# shuffle our features and turn into np.array

random.shuffle(training)

training = np.array(training)

# create train and test lists. X - patterns, Y - intents

train\_x = list(training[:,0])

train\_y = list(training[:,1])

print("Training data created")

from tensorflow.python.framework import ops

ops.reset\_default\_graph()

# Create model - 3 layers. First layer 128 neurons, second layer 64 neurons and 3rd output layer contains number of neurons

# equal to number of intents to predict output intent with softmax

model = Sequential()

model.add(Dense(128, input\_shape=(len(train\_x[0]),), activation='relu'))

model.add(Dropout(0.5))

model.add(Dense(64, activation='relu'))

model.add(Dropout(0.5))

model.add(Dense(len(train\_y[0]), activation='softmax'))

# Compile model. Stochastic gradient descent with Nesterov accelerated gradient gives good results for this model

sgd = SGD(lr=0.01, decay=1e-6, momentum=0.9, nesterov=True)

model.compile(loss='categorical\_crossentropy', optimizer=sgd, metrics=['accuracy'])

#fitting and saving the model

hist = model.fit(np.array(train\_x), np.array(train\_y), epochs=200, batch\_size=5, verbose=1)

model.save('chatbot\_model.h5', hist)

print("model created")

intents = json.loads(open('/content/intents.json').read())

words = pickle.load(open('/content/words.pkl','rb'))

classes = pickle.load(open('/content/classes.pkl','rb'))

# Utility Methods

def clean\_up\_sentence(sentence):

    # tokenize the pattern - split words into array

    sentence\_words = nltk.word\_tokenize(sentence)

    # stem each word - create short form for word

    sentence\_words = [lemmatizer.lemmatize(word.lower()) for word in sentence\_words]

    return sentence\_words

# return bag of words array: 0 or 1 for each word in the bag that exists in the sentence

def bow(sentence, words, show\_details=True):

# tokenize the pattern

    sentence\_words = clean\_up\_sentence(sentence)

 # bag of words - matrix of N words, vocabulary matrix

    bag = [0]\*len(words)

    for s in sentence\_words:

        for i,w in enumerate(words):

            if w == s:

                # assign 1 if current word is in the vocabulary position

                bag[i] = 1

                if show\_details:

                    print ("found in bag: %s" % w)

    return(np.array(bag))

def predict\_class(sentence, model):

# filter out predictions below a threshold

    p = bow(sentence, words,show\_details=False)

    res = model.predict(np.array([p]))[0]

    ERROR\_THRESHOLD = 0.25

    results = [[i,r] for i,r in enumerate(res) if r>ERROR\_THRESHOLD]

# sort by strength of probability

    results.sort(key=lambda x: x[1], reverse=True)

    return\_list = []

    for r in results:

        return\_list.append({"intent": classes[r[0]], "probability": str(r[1])})

    return return\_list

def getResponse(ints, intents\_json):

    tag = ints[0]['intent']

    list\_of\_intents = intents\_json['intents']

    for i in list\_of\_intents:

        if(i['tag']== tag):

            result = random.choice(i['responses'])

            break

    return result

def chatbot\_response(text):

    ints = predict\_class(text, model)

    res = getResponse(ints, intents)

    return res

res = chatbot\_response('what is FAT exam ')

print(res)