International Journal of Engineering Sciences & Emerging Technologies, Oct. 2023.
ISSN: 22316604 Volume 11, Issue 2, pp: 48-54 ©IJESET

SMART CHATBOT USING NLP

Neha Gupta, Amay Bhatnagar, Hayat Mehmood Usmani

Department of Computer Science & Engineering, MIT Moradabad, Moradabad, India
discoverneha@gmail.com
amaybhatnagar13@gmail.com

ABSTRACT

A chatbot (formerly chatterbot) is a software application that aims to copy human conversation through text or voice interactions. Modern chatbots are artificial intelligence (AI) systems that are capable of maintaining a conversation with a user in natural language. Such technologies often utilize aspects of deep learning and natural language processing. Sometimes due to lack of resources human interaction is not possible thus chatbot is made just for that [1]. As it can be available anytime and anywhere a use wants. As it is an AI-powered machine it can perform multiple tasks together. NLP or Natural language Processing is a tool that is responsible to interpret and answers in human language .The ultimate goal of this project is to develop a chatbot that will tell you about the working and history of the program while interacting and teaching about the chatbot to the user at the same time using various machine learning techniques and algorithms.

KEYWORDS: *AI-Powered Chatbot, NLP, Machine learning algorithms, Human language*

1. INTRODUCTION

Chatbot is now one of the most trending piece of software program that is available in the market currently. As it is used in interacting and giving the user commands that the machine that execute using the interpretation made by the chatbot itself. Google assistant, Amazon Alexa, Apple siri are some of the most advance and sophisticated AI-powered chatbots that are made. These are very helpful as they are used in day-to-day task. They work on internet thus they constantly receive latest updates and new features provided by the company thus improving the user interaction and performance of the machine. The chatbot is trained using various Machine learning and AI based algorithms and techniques to make it work optimally. NLP is most widely used technique to make the Chabot as it help to understand the language and sentences just like humans do. Chabots can run on both offline and online mode. In online mode the user have an interface and the Chatbot gets its dataset from the cloud, example: - Wikipedia, citizendium etc. Whereas offline chatbots depends completely on the dataset and information that available on the same servers and answers to a limited extent only.

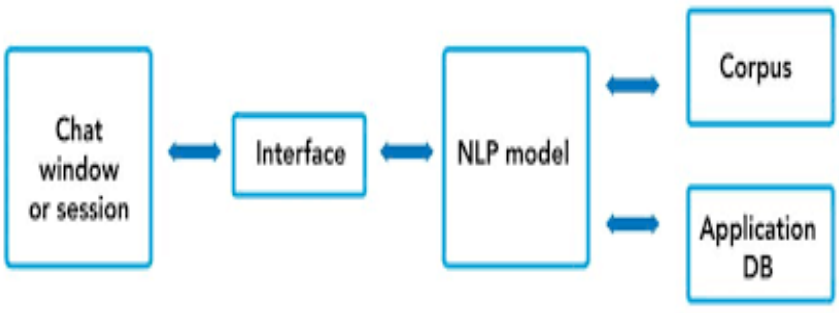


Figure 1. Architecture of the Chatbot

DOI: 10.5281/zenodo.10434221

2. LITERATURE REVIEW

Chatbots have emerged as a prominent and versatile technology, garnering increasing attention in various domains due to their potential to engage users in natural language conversations. Chatbots have been extensively explored for their applications in customer service, healthcare, and education. Research indicates that chatbots can enhance customer support by providing quick and efficient responses, leading to improved customer satisfaction and reduced operational costs for businesses[2]. Moreover, in the healthcare sector, chatbots have shown promise in offering medical information, mental health support, and personalized assistance to patients, augmenting healthcare services and accessibility. Additionally, in the education domain, chatbots have been utilized to deliver interactive learning experiences, answering student queries and providing timely feedback and is being deployed by leading coaching and educational institutes even in a large scale even in developing countries like India. Despite these promising advancements, some researchers have highlighted concerns surrounding data privacy, potential biases, when interacting with chatbots. Further research is needed to address these challenges and explore new avenues for chatbot development, leading to more sophisticated and empathetic chatbot systems that cater to diverse user need

3. SYSTEM IMPLEMENTATION

This chatbot program is based offline that gives response to user queries. The architecture of the chatbot is shown below in Figure 1.The user is first greeted by the chatbot then the user is allowed to ask the query where the user types his/her required query where the chatbot then responds back with an appropriate answer. If the user is unsatisfied with the response then the user can continue by briefly elaborating its queries even more.

3.1 Importing Libraries

Library are the important part of python programs as it help us to use various function that are reduce our coding time and the lines of code. Thus to make a chatbot we import various python libraries such as io ,strings ,warning ,numpy ,sklearn ,NLTK.

```
import io
import random
import string # to process standard python strings
import warnings
import numpy as np
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity
import warnings
warnings.filterwarnings('ignore')

import nltk
from nltk.stem import WordNetLemmatizer
nltk.download('popular', quiet=True) # for downloading packages
```

Figure 2. Importing libraries

For NLTK package to be install internet connectivity is required for the first time. Thus system install punkt and wordnet for one time only.

DOI: 10.5281/zenodo.10434221

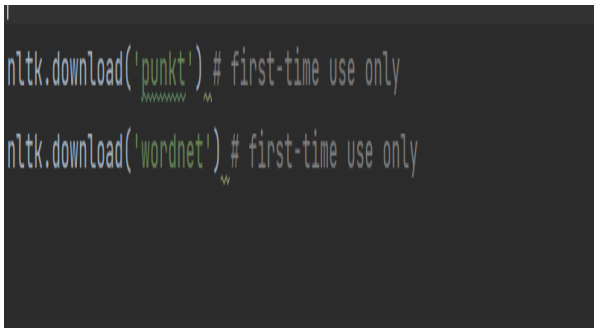


Figure 3. Importing and Downloading Punkt & Wordnet

3.3 Reading Database

Making a chorus about the Chabot history and its usage and converting it into a .txt. As it is a simple text file so it need to be preprocessed before NLP can be implemented on this text file using various techniques. Using lower() function we convert the uppercase letters to lowercase to simplify the text easily.

Figure 4. Getting Corpus txt. File

3.4 Tokenization

Tokenization is just the term used to describe the process of converting the normal text strings into a list of tokens i.e words that we actually want. Sentence Tokenizer can be used to find the list of sentences and Word Tokenizer can be used to find the list of words in strings[3].

An application database in the context of a chatbot refers to a structured collection of data that the chatbot uses to store and retrieve information related to its specific functionalities or domain. The application database serves as a memory for the chatbot, allowing it to maintain context, remember user preferences, and provide personalized responses. It include various information that were collected using the execution of the program such as transaction history, user information, sentiment analysis etc.

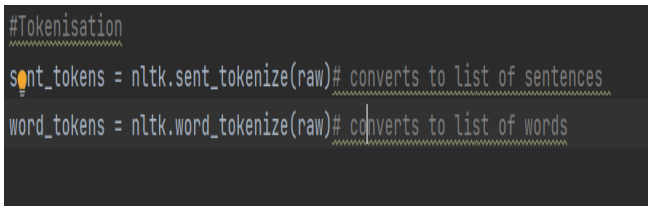


Figure 5. Conversion to tokens

3.5 Preprocessing and Lemmitization

The Process of reducing the different forms of a word to one single form and process of grouping inflected forms together as single base form.

DOI: 10.5281/zenodo.10434221

```
lemmer = 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)))
```

Figure 6. Preprocessing and lemmatization of tokens

3.6 Keyword Matching and Generating Response

Now here we define the user greeting for the bot and the response the bot gives back. Here we use simple word matching for generating user greetings to get back simple greeting randomly from the set of greeting data

```
# Keyword Matching
GREETING_INPUTS = ("hello", "hi", "greetings", "sup", "what's up","hey",)
GREETING_RESPONSES = ["hi", "hey", "*nods*", "hi there", "hello", "I am glad! You are talking to me"]

def greeting(sentence):
    """If user's input is a greeting, return a greeting response"""
    for word in sentence.split():
        if word.lower() in GREETING_INPUTS:
            return random.choice(GREETING_RESPONSES)
```

Figure 7. Generating words by keyword matching

3.7 Generating Response

At this section we will be generating our responses for our chatbot using the techniques term frequency[tf] which decide the most optimal response for our chatbot by calculating the most reused words in the sentence from the given dataset thus giving the most close and fairly accurate response .We will also implement the Cosine similarity as it help us to measure the similarity between the two support vectors and help us determine whether the two vectors are in same direction or not thus helping us in document text analysis[4].
So for generation of the response the concept of document similarity will be used where when the user gives the input the bot searches the most appropriate response the given text document to return one of the possible values. If it don't find anything it will return a response: "I am Sorry, I don't understand you!"

DOI: 10.5281/zenodo.10434221

```
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
```

Figure 8. Using Term Frequency to get response

3.8 Initiation and Ending Conversation

Now at this step we will define the initiation and the ending conversation for our chatbot. This function will define the welcome and ending line of the chatbot and we will also the bind the word ‘bye’ to the exit of the chatbot close the chatbot stop automatically. These initiation and ending stanzas we help the user to know whether the chatbot is ready to use or not.

```
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..")
```

Figure 9 . Setting of start and stop responses of the chatbot

3.9 Getting the Output and Working!

At this stage we will finally test our chatbot and check whether its gives back the required response or not. The Chatbot will run at the terminal of our IDE. First it will load the required NLTK library and tools from the internet and if already present it will inform for the same.

DOI: 10.5281/zenodo.10434221

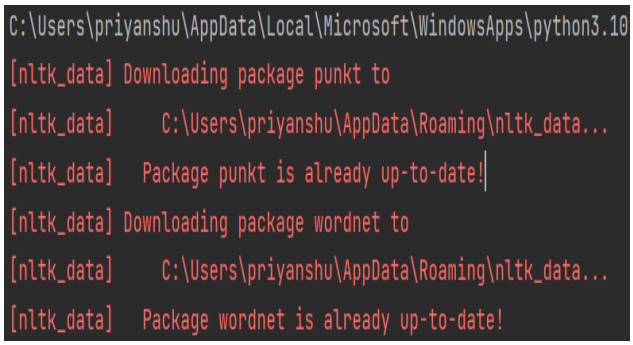


Figure 10. Downloading NLTK package

As it is already downloaded thus it will gives the message for the same. As all the required libraries are finally loaded thus we will are now ready to use our chatbot and to process with our required queries. The chatbot we will greet us with the simple greeting message and thus we can now proceed.

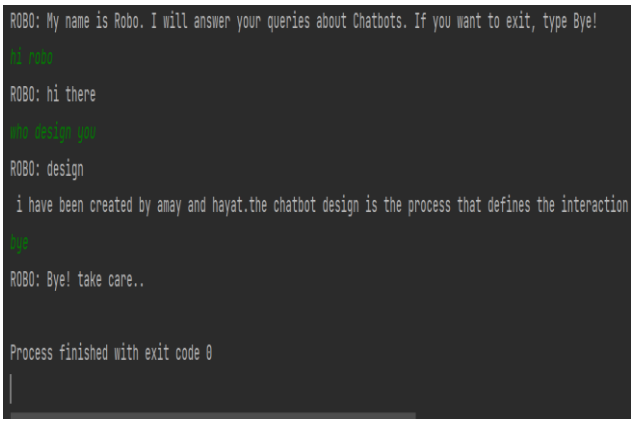


Figure 11. Starting conversation with chatbot

Thus we can clearly by responding the word ‘bye’ the chatbot closes its operation automatically. We can also see that our chatbot give us the required output that we want.

4. CONCLUSION

As the program is executed without any errors thus we can say that the program is running successfully. We have made this college specific chabot named “ROBO” that is light weight and is very user friendly. This chatbot will help the user to ask the queries about the working of chatbot and its history. Through the use of various deep learning techniques like NLTK, tokenization, stemming, wordnet etc. We were able to implement a domain specific chatbot that cater to our needs and was running successfully. Through various testing and evaluation the Chabot validated its performance.

The chatbot runs on the terminal window thus no extra GUI window is required thus decreasing the overall time index of the program. This program validates that using various tools and python libraries we can develop a very efficient chatbot program that can perform various types of inputs and can give most accurate answers by searching from the dataset given. The Project also demonstrated the importance of various data cleaning, data mining and data preprocessing techniques such as tokenization, wordnet, stemming etc.

Furthermore it highlighted the significance of NLP as powerful and comprehensive tool that help us understand and implement human language in a computer system with use of various techniques and

DOI: 10.5281/zenodo.10434221

algorithms. The Chatbot comes with high level of customization and thus many features can be added later to get vast amount of data if necessary by increasing the size of its dataset respectively. By Including other APIs and dataset of sports, news and weather services and other it can also answers these related questions outside of its dataset and thus increasing its operation capabilities [5]

It may be noted that the output by the chatbot may wary as it uses Natural Language Processing thus it keep learning more and more as the time passes by and thus increasing its accuracy even more.

In Conclusion the chatbot program demonstrated the advancement, effectiveness and potency of various AI and ML techniques such as NLTK, Sk-Learn and other various libraries that can be used to make complex and sophisticated applications that can contribute to various branches of ai based computer systems.

REFERENCES

[1] http://en.wikipedia.org/wiki/Chatterbot

[2] https://www.ijraset.com/research-paper/chatbot-using-python

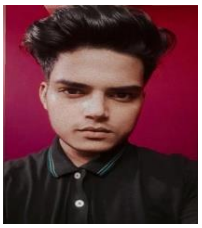
[3] https://towardsdatascience.com/tagged/chatbots

[4] www.geekforgeeks.com

[5] Bhaumik Kohli , Tanupriya Choudhury, Shilpi Sharma, Praveen Kumar., A Platform for Human- Chat bot Interaction Using Python, IEEE , 2018

AUTHORS

Neha Gupta is an assistant professor in Computer Science and Engineering Department of Moradabad Institute of Technology affiliated with Dr. A.P.J. Abdul Kalam Technical University. She had done B. Tech, M.Tech and now pursuing Ph. D. Her research area involves Machine learning, Deep Learning, Data Science and Data Security Measures.



Amay Bhatnagar is a B.Tech 3rd Year Student in Computer Science and Engineering Department of Moradabad Institute of Technology affiliated with Dr. A.P.J. Abdul Kalam Technical University. His research interests include Artificial Intelligence, Machine Learning and Natural language Processing.

.

Hayat Mehmood Usmani is a B.Tech 3rd Year Student in Computer Science and Engineering Department of Moradabad Institute of Technology affiliated with Dr. A.P.J. Abdul Kalam Technical University. Her research interests include Artificial Intelligence, Machine Learning and Natural language Processing.

DOI: 10.5281/zenodo.10434221