



Artificial Intelligence based Chatbot: A Case Study

Nidhi Singh Kushwaha¹, Pawan Singh²

^{1,2}Amity School of Engineering and technology, Amity University Uttar Pradesh, Lucknow Campus, India
nidhikush2016@gmail.com¹, pawansingh51279@gmail.com²

How to cite this paper: N. S. Kushwaha and P. Singh, "Artificial Intelligence based Chatbot: A Case Study," Journal of Management and Service Science, Vol. 02, Iss. 01, S. No. 004, pp. 1–13, 2022.

<http://doi.org/10.54060/JMSS/002.01.004>

Received: 01/01/2022

Accepted: 15/02/2022

Published: 25/02/2022

Copyright © 2022 The Author(s).

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0>



Open Access

Abstract

This paper shows the implementation of an artificial intelligent chatterbot with human interaction. Our paper will now tap into the aspects of artificial intelligence because we are designing a smart chatbot and of course, using machine learning to do so and deep learning which is involving more in-depth concepts of artificial intelligence and tapping into the territory of unsupervised learning from data more with the help of neural networks and more complicated models which help with studying better the unstructured or the unlabeled form of data. The name of our Chabot will be finance manager. It will interact with the user efficiently and come up with reasonable responses for their queries, we will also have a module to help people with their wellness in our chat bot apart from it just being a plain financial aspect talker for a more refined personal touch to our paper.

Keywords

Chatbot, Artificial Intelligence, Natural Language Processing

1. Introduction

Many national and international, multinational companies are basically focusing on their own chatbot to make it easier for their customers and for better, easier business. A chatbot is basically an artificial intelligence system which can do conversation with human beings. This could be text-based, or a spoken conversation (in case of voice-based queries). Chatbots are basically used to get information instantly without contacting to human. It can run on your personal computers and mobile phones too. Although most of the times it can be accessed only through the internet. It is basically a conversational agent which basically interacts with user in specific domains by receiving input in Natural Language Sentences. It can also be referred as software agents. A chatbot is a program that speaks with you. It is a layer on top of, or a passage to, a help. Now and again it is controlled by AI (the chatbot gets more brilliant the more you associate with it). Or then again, more generally, it is driven utilizing savvy rules (for example assuming the individual says this, answer with that). The administrations a chatbot can convey are different. Significant life-saving wellbeing messages, to really look at the weather conditions estimate or to buy another set of shoes, and whatever in the middle between. The term chatbot is inseparable from message discussion

however is developing rapidly through voice correspondence... "Alexa, what time is it?" (other voice-chatbots are accessible!)

The chatbot can con-verse with you through various stations; like Facebook Messenger, Siri, WeChat, Telegram, SMS, Slack, Skype and numerous others. Purchasers invest bunches of energy utilizing informing applications (more than they spend via virtual entertainment). Along these lines, informing applications are right now the most famous way organizations convey chatbot encounters to shoppers

2. Related Work

NLP is a field/combination of various aspects of machine learning collectively used together which included in it – computer sciences, artificial intelligence and linguists mainly concentrated in the communications and interactions between the computers and human languages but these buzzword technology are much more intricate and complicated in their implementation than they seem to be ,the communications and interactions require for perfect sense and understanding of all aspects between user and the system which is hard to get right and implement perfectly , given its mode of communication because one word can have multiple meanings and connotations attached with it because human beings , when we speak , we have a mood / feeling attached to a word depending upon the tone in which it was spoken for example – a word as simple as “what” – can be used in multiple scenarios to indicate or express different emotions . Like, “what” can be a person asking another about something or “what” can also indicate disbelief or shock related to as the response of an event like as if can this even be true. So, the NLP toolkit is used for ambiguity disintegration between words, sentences – handling tokenization, handling POS, which is parts of speech tagging, which stands for parts of speech tag based on its context and definition. It’s very useful in sense disambiguation. In seeing the word in its true content sense and the function of the word as in if it is an interjection, pronoun, conjunction, preposition or a verb, noun, adjective or adverb. Words in natural languages are polysemous. In the lexical, that is, the word-to-word sense, there are – synonyms, antonyms, gradation and in the semantic sense, hypernymy, entailment, meronymy or holonymy and toponymy.

3. Proposed System

To use recurrent neural networks, you need to additionally install the Keras and the Tensor flow library by using pip install commands. To start with, the name shows the functionality, a chat bot is like a robot who chats with you. It’s a program/software that will have interactions with you and its of two varieties –

- The first kind, a rule – based chat bot, which is more of a guided – kind of chat bot. It has rule-based learning, it answers questions on the bases of the rules its being trained on. It does not shift from the things it already knows.
- The second type is a self- learning chatbot which is more independent, free and, flexible in its methods and learning. We will be focusing more on the self – learning aspect of our chatbot primarily. First, let’s dive into Nlp, which is natural language processing, which will greatly help us with our chatbot designing. We have to install NLTK first, to make use of all nlp modules of python. We will also involve sentiment analysis. A chatbot automatically gives response or generates answers to the questions or queries of the user. It’s like an informal informer, instead of talking to a person, it can do the same. Siri, alexa are AI smart voice assistants that are intelligent, interactive, advanced.

The computer only understands the language of 0’s and 1’s. So, the process of us trying to get it to under-stand smartly, the user’s sentiments, emotions and meaning can be tough. In nlp, most models which are basic in nature are based on the concept of “bag of words” but the problem with it is that it does not capture the syntactic relations or any actual meaning out of the sentences because the “bag of words” only maintains the count or multiplicity of the words in a given text or arti-

cle. for example – “what it feels like is it will not like a snake”, so the bag of words will contain words and count of words does not making any easier the actual task at hand. It disregards any real actual meaning or grammar of words. Any nlp algorithm or program, does not directly work on our text, works on numbers actually. So, our bag of words, keeps count of most frequently used words and stop words – that consists of the mostly used generic and basic words used to complete sentences commonly but Sentiment Analysis with it, with complete accuracy is not possible. Like in the following sentences– “I like you” or “I look like you” are completely different sentimental use of the word like – in the first sentence, the emotion is positive and in the second one , the emotion is neutral , just stating a fact or opinion ,a plain statement . The first step is data gathering which can be done with Spacy – a nlp mod-ule with structured data with many build – in features containing many human languages. Other libraries that can be used are beautiful soup, Selenium.

D3.js is also used which is a java script library used when the number of customized features is in. For re-solving lexical ambiguity at word level, syntactical ambiguity at the sentence level and referential ambiguity at the referring level. Also, eliminating tags by using stemming and lemmatization. It is very important to train the chatbot we made because that is the way it can improve itself, by saving the user input and corresponding responses. The greater the domain – knowledge, the better the chatbot designed is. The data cleaning and transformation is with pandas, numpy and spacy. for the purpose of visualization comes the matplotlib, plotly.

If we look forward to using our chatbot in some web-based application django, flask are the libraries to go for. OpenCV, Librosa, scikit learn, pytorch, tensorflow for advance level analytics and audio, image recognition. For further use, it is also possible to integrate our created chat bot into other environments where it will be more accessible to the user like in a webpage, website or through the use of an app

4. Design & Architecture System

Use Case Diagram:

(User and admin roles)

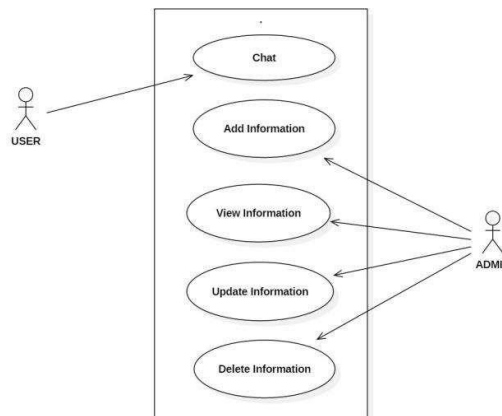


Figure 1. Use case diagram of user and admin roles

Data Flow Diagram:

Level 0:

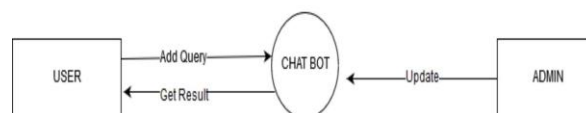


Figure 2. Zero level DFD of Chatbot system



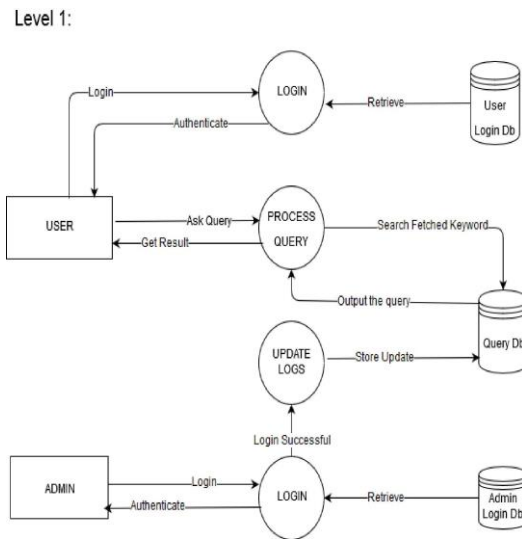


Figure 3. Use case diagram of context identification

Activity Diagram:

Personal Query Response Activity (Module-1):

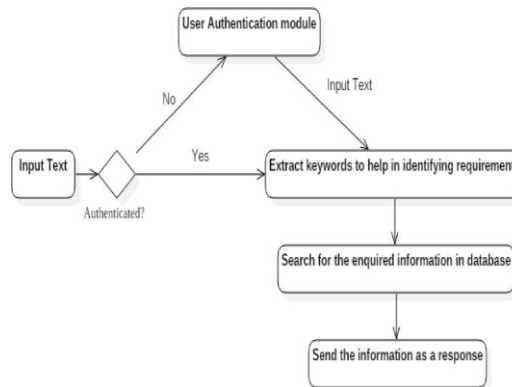


Figure 4. Activity Diagram of Personal Query Response Activity

Normal Conversation Response Activity (Module 2):

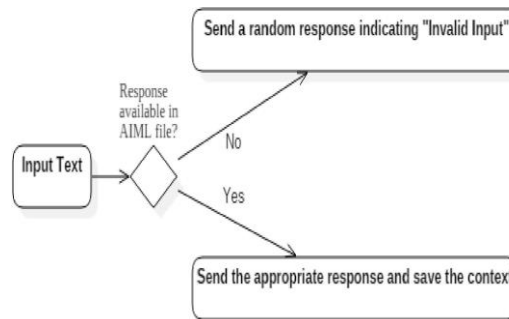


Figure 5. Activity Diagram of Normal Conversation Response

• Conversational Chatbot Architecture

Usually, chatbots exclusively upheld a solitary nearness pair, otherwise called a single shot discussion. Notwithstanding, current chatbots can support numerous nearness matches, recalling states and set-tings among discussions and have the ability to relate information in various contiguousness matches which is connected. This is the bot's capacity to protect the discussion. A chatbot comprises of four principal parts: front-end, information base, back-end and corpus which is the preparation information. The front end is responsible for empowering correspondence between the bot and the client. The NLU uses artificial knowledge strategies to distinguish the aim and setting of the client input. A suitable reaction is produced from the clients' expectation. The information base characterizes the chatbots information, which is made inside the NLU and upheld by the back-end, the back-end applies the spaces corpus to de-liver the information base.

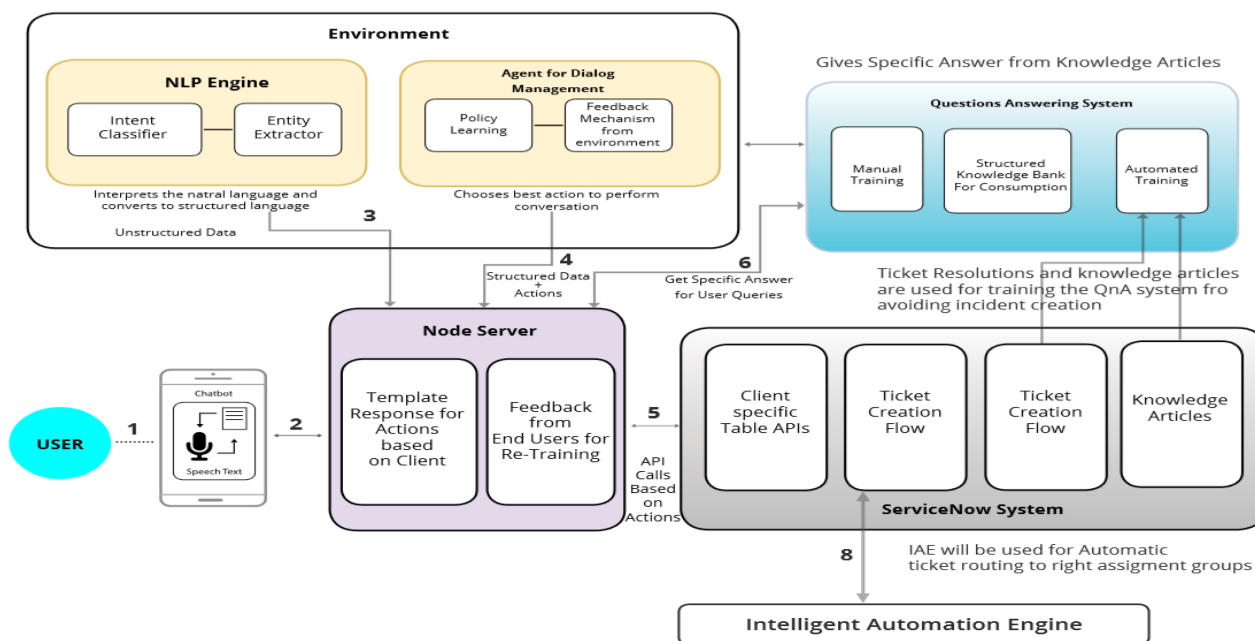


Figure 6. Architecture of Conversational Chatbot

5. Methodology

So, first we will be working with the 3.6 version of python in our paper. We use the command -

```
conda create -n chatbot python=3.6
```

Chatbot can be replaced with whatever name you want for your chatbot, can be sentiment analyzer, financial master etc. activate chatbot

This command activates our chatbot. Then, the most important library,

- Pip install nltk
- pip install tflearn
- pip install tensorflow

Now, that we have all the essential libraries we will need installed – deep learning, natural language processing for analysis and more efficient chatbot, we start working on out jupyter or pycharm environment –

- import nltk



- from nltk.stem.lancaster import LancasterStemmer

By importing the Lancaster stemmer, we will be able to perform lemmatization or stemming which helps us get the word to its root form like – teacher, teaching, teaches are all coming from the root word teach, so, it makes our processing easier.

After this, we import the libraries - numpy, json, tflearn , tensorflow and random Textblob, genism.

The first chatbot was called ELIZA developed in 1964. It was fairly simple and not complicated, it used to assign a numerical value to every word in the given sentence with the most significant word given the highest value and then frame a question accordingly. It had limited capacities, but it did not over – complicate things. Chat bots today are fairly much more advanced capable of having human– like interactions. Various companies use it to reduce manpower.

Chatbot uses nlp which is connected to a knowledge base and data store in nlp. Text or speech is converted to structured data by sentiment analysis, tokenization, normalization, dependency parsing. The intelligence of a chat bot depends on how appropriately it responds using complex algorithms, mapping the input into useful representation. The problem is that it suffers with the inherent complexities in the human speech like similes, metaphors, so, it can get hard sometimes to correctly understand the conversations. Volume of conversations, training of agents, consistency of agents, length of conversation, complexity of incoming and outgoing questions helps to improve chatbot. The chatbot has layers. The first is the nlp layer which is all about the components of artificial intelligence dealing with languages and the part with natural language understanding or nlu. Firstly, mapping the input into useful representation because nlp can process only numbers and analyzing the different aspects of knowledge - text planning, sentence planning, text realization.

Second is the neural networks which is as we discussed, information processing paradigm inspired by brain It extracts meaning from complicated, imprecise data and extracts patterns and trends. Deep learning is the third layer dealing with unstructured, unlabelled data inspired by ANN (artificial neural network). It understands a problem and if deep learning is a good fit or not, identify relevant data, preparing for analysis, building analytical model based on chosen algorithm, training algorithm on large amount of labeled data, tests against unlabelled data. It works in an incremental manner to learn high level features. It doesn't require domain expertise and hardcore feature extraction.

The fourth layer is tensor flow, which is extremely important as machine learning is really intricate and complicated but with this library, the implementation of these models becomes very easy. It is for numerical computation , large scale machine learning bundles together a slew of machine learning and deep learning models and algorithm makes them useful by way of a common metaphor .In designing our own personalized chatbot , we define the role of the chatbot first, as in , what is the purpose of the chatbot actually , the conversational architecture, dialogue flow , question variations that can occur , picking the platform and development approach , NLP (dialogue flow , engineering NLU) , internal testing and use case detection .

Human language is a mystery for the computer. There are over six thousand five hundred languages over the world, so, text mining which is deriving meaningful information from natural language text. Now, we have already installed all essential libraries. We have imported Lancaster Stemmer which is for our text normalization to convert our data into inflexion, that is, common root word form. Tokenization, stemming, lemmatization, POS tags, named entity recognition, chunking are all of the techniques we implement for text normalization. Stemming is faster than lemmatization, it does not essentially require POS tagging and unlike lemmatization, does not use word Net corpus.

Lexical knowledge networks, Word net theory, word nets made and designed for specially Indian languages, multilingual dictionaries, semantic roles, word sense disambiguation, multilingualism, metaphors, co references all help to classify and differentiate in what the user is actually saying , involving machines or robots to understand and process the language that hu-

mans speak at the lexical or word level , syntactical or parsing or referential level too .

Knowledge network includes domain addressed by the network, principles of construction, methods of construction, representation, quality of database, applications, usability mechanisms for software applications and users – application programming interfaces (APIs) , record structures , size .We are using Lancaster stemmer because it is more aggressive . Out of the two stemmers available for English language, the other being Poder, it's the more advanced one. Porter stemmer uses suffix stripping to produce steps. It does not follow linguists, its simple and speedy using a look up table and fast retrieval. For better and simpler implementation, we can also install chatterbot directly,

- pip install chatterbot
- from chatterbot. trainers import ListTrainer
- chat= ["HELLO!",
- "hi",
- "How are you doing these days?", "I'm doing good."
- "Nice talking to you", "Same here"]

The chat part can include all the common conversational courtesies. In our case, we are building a finance and mental health sentiment analysis chat bot, so, we can load our finance related detailed conversational data and our basic mental health awareness data. The key is that our chat bot cannot be perfect in all its responses, it cannot respond to all random queries till we train it with a very large amount of data. The data science workflow with natural language processing goes as , we start with a question (asked by the user asking to the chat bot) , then we get the clean data by all the techniques we learned for better understanding of our data and the actual sentiment of the users after this , we perform the third step , which is performing explanatory data analysis , applying techniques and then after effectively understanding the question , going through the data , share its in-sights with the user . We collect the most common or most likely to be asked questions. We create intention groups. Our intention group will be separate and contain the most likely to be asked questions together. For example –

Intention 1 can have questions about the basic terms and meanings of things in the stock market and terminologies related with finances, like what is the stock market, who controls it, is it open on all days, what is the impact of the ongoing covid – 19 on the stock market etc. The second intention can be about more general things and mental health issues, because we are creating a finance and better lifestyle with a positive mental health assignment so, the next intention could be about the basic symptoms of mental health diseases like anxiety, bipolar or eating disorders etc., greetings. The first part is about Programming – dealing with data and NLP which can be done with implementing several useful libraries like pandas , sklearn, nltk, Next is related to math and stats – clean the data , (corpus , document) EDA (which stands for Explanatory data analysis) for word counts, Nlp for sentiment analysis , topic modeling , text generation .Final stage of data science workflow is design and domain which includes scope , visualization , extract insights expertise . So, the order for our NLP and making chat bot goes as the following –

1. Question
2. Data Gathering, cleaning
3. EDA
4. Techniques
5. Insights



- from nltk. chat. util import Chat, reflections

Reflections is a kind of dictionary which contains a set of input values and the most probable corresponding output values. In the data gathering category, try to limit your scope to the most important and essential data.

- Web Scraping – It is a very essential step as data extraction is important for the type and performance of chat bot you are making. It converts the unstructured data of all kinds available in search engines, news sources, social media pages, websites by web harvesting to a structured form like json file, pickle, or spread sheet like format – XLS, csv, sql, xml. Two ways we do it is –
 - Requests – We make http requests or get info from a website
 - BeautifulSoup – parse html documents, extract parts of a website.

After scraping, save the data in a pickle file to serialize the python objects and save it for later. Then, goal is clean, standard format data for further analysis in the form of corpus, document – term matrix.

Corpus is just a collection of texts. We do this by using pandas to create data frame, which is data in the form of a table for example –

Then, we create a document – term matrix, which is a bit more complicated as we clean text removing the unnecessary parts, tokenize the text, put it into a matrix so that a machine can read it as nlp only reads numerical. So, tokenization – split text into tokens, smallest token can be a word or sentence. Over simplify, re-move stop words, bag of words (sequence does not matter). Each row is a different document or transcript, for example – each row can indicate one disease and each column a different term, word or can also be a bi-gram so it can be symptoms appearing and the values will indicate the word counts. The scikit-learn function Count Vectorizer is the way to make DTM which stands for document – term matrix.

6. Implementation

We do data scraping by running the following code on our desired data url for the transcript –

```
def get_transcript_from_url(url):
```

- pg=requests.get(url).text

#This will fetch for us all the text from the url, you can get details of the class containing the transcript by going through the inspect and then hovering the mouse over the class containing the text to get its actual name. sp=BeautifulSoup (pg," lxm1")

```
t= [d.text for d in sp.find(class_="#name of the class").find_all('d')] print(url)
```

```
return t
```

```
transscp= [get_transcript_from_url ( r) for r in urls] #urls is a list con
```

```
for i, j in enumerate(stocks):
```

```
with open("Transcript/" +j+".txt","wb")as file:
```

#Wb is a file mode in python that means / indicates that you are writing in a file and in binary mode.

```
pickle. dump(t[i],file)
```

```
#now, we will begin to load our pickle file- dataset= {}
```



```
for i, j in enumerate(stocks):
with open("Transcript/" +j+".txt","rb")as file: dataset[j]=pickle.load(file)
dataset. keys()
```

#Dataset is a dictionary that we have created in which each key signifies a disease and every value is a transcript of symptoms of the disease

Now, cleaning the data is the next step – make data in lower case, remove punctuations , numerical values , common non-essential parts of the text , tokenize the text , stop words should be removed . After tokenization, stemming / lemmatization , parts of speech tagging , creating bi- grams or tri – grams – all help in removing re-dundant data and keeping only most essential parts .

Now, we create a function or method that combines all the various chunks of paragraphs of data together which we gathered from various sources as text data combined together is a lot easier to deal with than various different files of data to look into separately . We use next () method to have a look at our data again for check –

```
next(iter ( dataset.keys ( )))
```

#next(iter (dataset. values ())) is used to iterate over our dictionary or files. This is for transcript values.

```
def combo_all(list_all):
combo_t = “.join(list_all) return combo_t
```

So, combining our data-

```
data_all ={key : {combo_t(value)} for { key , value } in dataset.items()}m Now , that we have combined all the various text data we collected from various urls we can either keep it in a dictionary format or in the form of a data frame . import pandas as pd
pd.set_option('max_colwidth',160)
df= pd. DataFrame.from_dict(data_all).transpose() df.columns=['transscp']
df=df. sort_index() df
```

Then, we further clean the data by removing the stop words , removing non important parts of the text we import re , string modules for this re is for regular expression , then returning the clean text and organizing it in the form of corpus and document – term matrix by using sklearn count vectorizer also pickling it for future use . First, instantiate a count vectorizer object , put in stop words (English) stop words as objects, then it automatically gets that we do not want them as individual columns .

```
Ob= CountVectorizer (stop_words = “English”)
```

```
# we can specify any language based on our chat bot interaction language.
```

It is very important to have a good, detailed and dependable dataset as the most basic principle of computer science which every starter in the field knows too is the principle of GIGO which is garbage in garbage out , so for our chat bot to be a good one and perform well ,it’s extremely important that the data we collected and are doing analysis on is good . Then we proceed to EDA. So, we need to take out top words (the most amount of words used) , the general vocabulary and terminologies which we can easily do with our document data matrix format , we just need to aggregate the data and even use

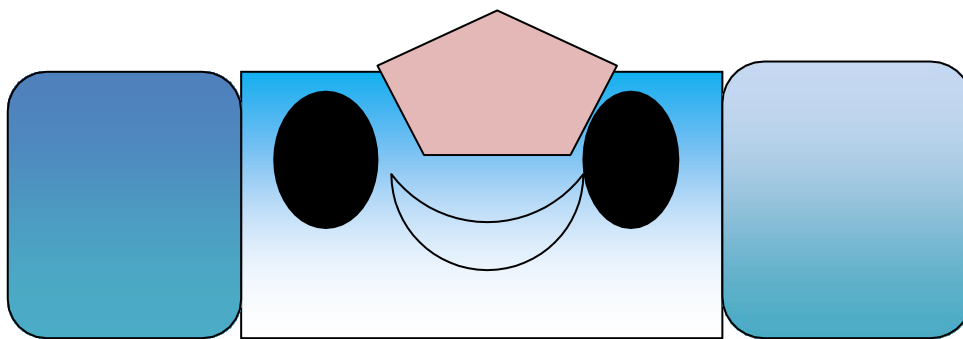
and create word clouds , bar plots or in case of just two words only , the scatter plot using the matplotlib package. The word cloud is more interesting but it requires an additional conda library to be installed. If your analysis still doesn't make sense, you need to do further cleaning, more . After this, the good thing about pandas is that it's very easy to read the pickled files again using the read_pickle function of pandas for data frame. Sentiment analysis is the next step and document matrix is not of much credibility or importance here because order does matter for classification as positive, negative or neutral emotions . We have TextBlob here, which is a library in python build over rather on top of the nltk which lets us do some rule based sentiment scores and it's easy to use . In TextBlob, a great linguist , Tom De smedt, he studied all the words in the English language and af-ter that , he manually labeled them as positive sentiment , negative sentiment and created a database .So, it gives us the polarity , sense and subjectivity of those words . It just is very basic and takes the average of the sentiment, thus , a rule – based approach for sentiment analysis . Statistical method like naïve bayes can also be used which assumes all features to be independent of each other.

```
from textblob import Textblob TextBlob (“# our statement “).sentiment
```

This is how we can use NLP to help with our model. Now, we had imported chat and reflections , we define chats for our chat bot like instead of just specifying just limited chat . For example – if we specifically provide our chat bot with the conversation bit ‘my name is Martha ‘ , it will respond with an appropriate reply only when the name specified by the user is Martha so , we use regular expression instead so that the chat bot does not require every time to specify a particular case only and other cases are unknown to it .[8], [9],[10].

Chat bots are the technology really useful for these times when so many startups are emerging in the global scene in order to reduce the resources on man power, you can do most of your job including ordering of prod-ucts , interaction with customers , chat with people to resolve their query with the advanced smart chat bot , marketing , setting appointments . Working with natural language processing and neural networks, as we go on feeding more information to our chat bot , we can make it better and more intelligent too .

```
convo = [  
[ 'my name is ( , *) ' , [ ' hi % 1 ' ] ] ,  
]  
So, now we name your chat bot and give it multiple conversations Finance_master = Chat ( convo , re-flections )  
Chat. _ substitute ( ' you are welcome to chat with me ' ) # chat . converse ()  
You can create your own dictionary. from chat-terbot import Chatbot  
from chatterbot. trainers import ListTrainer convo = [  
"hello " , " Bonjour " , " How are you doing these days " , " I am good"  
]  
trainer = ListTrainer ( chatbot )  
# the chatbot references to our bot trainer. train (convo)  
To get a response,  
res = chatbot. get_response ( "Good day to you ! Switch on the inner sunshine " )  
print (res}
```



```

We use the tkinter gui of python to design our chatbot. main_obj=TK ()
main_obj. geometry ("600* 750") main_obj. ti-tle ("Finance manager ")
imag = PhotoImage (file = "#name of your flie" )
photoLb = Label (main_obj , image = imag ) # we have created an object of the label class
photoLb. pack (pady = 5) frame = Frame (main_obj ) scr = Scrollbar ( frame)
mgs = Listbox (frame, width = 80 , height = 20) scr.pack(side= LEFT , fill = Y)
mgs. Pack (side = RIGHT , fill = BOTH , pady = 12) frame. pack ()
textF = Entry (main_obj , font = ("Algerian" , 20)) textF.pack(fill = X , pady = 10)
button = Button(main ,text=" Ask Lifestyle master" , font=("arial black " , 20), command = ask_lifestyle_master )
button. pack ( )
def ask_lifestyle_master() :
print (" received " ) query = textF . get()
answer_lifestyle_master = bot. get_response(query) mgs.insert( END , " YOU : " + query )

#print(type ( answer_from_bot )) if you want to check the type it returns . mgs. insert ( END , " Fi-nance manager : " + an-
swer_lifestyle_master )

```

7. Result

The chatter bot performed up to the marked as in case of text provided, results produce were 100 percent accurate and more reliable. But in case of voice enabled input due difference in accent of user and Microsoft speech synthesizer model whose accent is based on UK or USA users. Thus, result obtain were close to the appropriate answer. Thus, from above it can be considered that speech synthesizer needs to be model for all kinds of accent such that it process the voice input more correctly, therefore chatter bot model can produce more accurate results.

8. Conclusion

In this paper, we learned about the implementation of the linear regression model of machine learning in python. Now we have an idea of machine learning and its basics-its types, the models included in it, its uses and applications, the various benefits it offer ,its modes of working ,error handling, concepts involved , functioning of a program in from tying it as the source code to the output while implementation of the model , machine learning and its uses spread across all fields . For imple-

menting the various machine learning models, we need to have proper knowledge of python, so now we can conclude that python is the programming language of the 21st century and has definitely helped in making programming easy in all aspects. Machine learning is a very deep and vast field with its uses and applications knowing no end. It can be so powerful when implemented in a judicious way which can be so resourceful. Python is the language of our generation. And with technology advancing and reaching new heights and simpler techniques every day, machine learning is going to be more useful for the coming generation and help make things easier, already being heavily used in various papers and purposes ranging from gaming to object detection and in the internet of things, robotics, smart homes etc. Machine learning is the future of technology, programming with new and better models and applications coming every day and simplifying things even further, the job is becoming easier and fun at the same time. In this paper, we successfully create for us a tkinter gui for our chat bot. Through this paper we have been able to understand and use natural language processing, nltk, python, machine learning in our conda environment to build an interactive chat bot application. The various build in modules, libraries and frameworks of python like pandas library, scikit learn library, chatterbot, chatterbot corpus, nltk were of judicious use for building our stock market prediction model with machine learning and chat bot using nlp, also included for the better understanding and accuracy of our model. The limitations remain that the chat bot has to be feed as much information, and that makes it smarter and more accurate, our chat bot is limited in its range , in order to make it smart lots of training is required , we can improve that by using neural networks and tensor flow libraries , chat bot is unable to understand human context and can't make decisions. It has the same answer for a query and no research skills or emotions. For future improvement, we can train our chat bot more and integrate it on a website.

References

- [1]. K.-J. Oh, D. Lee, B. Ko, and H.-J. Choi, "A chatbot for psychiatric counseling in mental healthcare service based on emotional dialogue analysis and sentence generation," in 2017 18th IEEE International Conference on Mobile Data Management (MDM), 2017, pp. 371–375.
- [2]. S. J. du Preez, M. Lall, and S. Sinha, "An intelligent web-based voice chat bot," in IEEE EUROCON 2009, 2009, pp. 386–391.
- [3]. B. Setiaji and F. W. Wibowo, "Chatbot using a knowledge in database: Human-to-machine conversation modeling," in 2016 7th International Conference on Intelligent Systems, Modelling and Simulation (ISMS), 2016, pp. 72–77.
- [4]. C.P. Shabariram, V. Srinath, C.S. Indhuja, Vidhya (2017). Ratatta: Chatbot Application Using Expert System, International Journal of Advanced Research in Computer Science and Software Engineering,2017
- [5]. Mrs Rashmi Dharwadkar¹, Dr.Mrs. Neeta A. Deshpande, A Medical ChatBot, International Journal of Computer Trends and Technology (IJCTT) Volume 60 Issue 1-June 2018
- [6]. Farheen Naaz, Farheen Siddiqui, modified n-gram based model for identifying and filtering near-duplicated documents detection, International Journal of Advanced Computational Engineering and Networking, ISSN: 2320-2106, Volume-5, Issue-10, Oct.-2017
- [7]. S. Kumar et al., "Novel method for safeguarding personal health record in cloud connection using deep learning models," Comput. Intell. Neurosci., vol. 2022, p. 3564436, 2022.
- [8]. S. Kumar, P. K. Srivastava, G. K. Srivastava, P. Singhal, D. Singh, and D. Goyal, "Chaos based image encryption security in cloud computing," J. Discrete Math. Sci. Cryptogr., pp. 1–11, 2022.
- [9]. S. Kumar et al., "Protecting location privacy in cloud services," J. Discrete Math. Sci. Cryptogr., pp. 1–10, 2022.
- [10]. Iop.org. [Online]. Available: <https://iopscience.iop.org/article/10.1149/10701.15533ecst/meta>. [Accessed: 23-Jul-2022].
- [11]. N-gram Accuracy Analysis in the Method of Chatbot Response, International Journal of Engineering & Technology. (2018)

- [12]. Aishwarya Hajare, Priyanka Bhosale, Rasika Nanaware, Guruswami Hiremath, Chatbot for education system ISSN:2454-132X, Impact factor: 4.295, Volume 4, Issue 3, IJARIT.
- [13]. Ardi, Hario Laskito. "Support Vector Machine Classifier for Sentiment Analysis of Feedback Marketplace with a Comparison Features at Aspect Level." (2017).
- [14]. H. L. Ardi, E. Sedyono, and R. Kusumaningrum, "Support Vector Machine classifier for sentiment analysis of feedback marketplace with a comparison features at aspect level," Core.ac.uk. [Online]. Available: <https://core.ac.uk/download/pdf/141497649.pdf>. [Accessed: 23-Jul-2022].