EASWARI ENGINEERING COLLEGE
RAMAPURAM, CHENNAI-89
AUTONOMOUS INSTITUTION

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

MINI PROJECT REPORT

191CSC504T – NATURAL LANGUAGE PROCESSING

Year/ Semester: III/ V Academic Year: 2021 -2022
(ODD SEM)

CHATBOT

A PROJECT REPORT

Submitted by

DHANWANTH KUMAR M

DHANUSH G

DILLI RANI R

ASHWATH B

SURYA M

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

EASWARI ENGINEERING COLLEGE

BHARATHI SALAI, RAMAPURAM, CHENNAI-89

AUTONOMOUS INSTITUTION

JANUARY 2022

EASWARI ENGINEERING COLLEGE

BHARATHI SALAI, RAMAPURAM, CHENNAI-89

(AUTONOMOUS INSTITUTION)

BONAFIDE CERTIFICATE

Certified that this project report **“CHAT BOT”** is the bonafide work of **DHANUSH, DHANWANTH, DILLIRANI, ASHWATH and SURYA** who carried out the project work under my supervision.

SIGNATURE **SIGNATURE**

HEAD OF THE DEPARTMENT **SUPERVISOR**

TABLE OF CONTENTS

S.NO	TITLE	PAGE NO.
	ABSTRACT	**6**
1.	**INTRODUCTION**	6
1.1	BACKGROUND	7
1.2	OBJECTIVES	8
1.3	PURPOSE, SCOPE AND	8
	APPLICAPIBLITY	
		10
2	**SURVEY OF TECHNOLOGIES**	
		11
3	**REQUIREMENT ANALYSIS**	11
3.1	PROBLEM DEFINITION	11
3.2	REQUIREMENT SPECIFICATION	13
3.3	SOFTWARE AND HARDWARE	
	REQUIREMENTS	14
3.4	PRODUCT DESCRIPTION	16
3.5	CONCEPTUAL MODEL	
		18
4	**SYSTEM DESIGN**	18
4.1	BASIC MODULES	18
4.2	DATA DESIGN	19
4.3	SCHEMA DESIGN	20
4.4	ALGORITHM	
		21
5	**IMPLEMENTATION AND TESTING**	21
5.1	IMPLEMENTATION APPROACHES	21

5.2	CODING DETAILS	23
5.3	TESTING	**25**
6	**RESULTS AND CONCLUSION**	25
6.1	RESULTS	26
6.2	CONCLUSION	27
6.3	FUTURE SCOPE	
7	**REFERENCE**	**27**

ABSTRACT

ChatBot can be described as software that can chat with people using artificial intelligence. These software are used to perform tasks such as quickly responding to users, informing them, helping to purchase products and providing better service to customers. In this paper, we present the general working principle and the basic concepts of artificial intelligence based chatbots and related concepts as well as their applications in various sectors such as telecommunication, banking, health, customer call centers and e-commerce. Additionally, the results of an example chabbot for donation service developed for telecommunication service provider are presented using the proposed architecture

Chatbot applications streamline interactions between people and services, enhancing customer experience. At the same time, they offer companies new opportunities to improve the customers engagement process and operational efficiency by reducing the typical cost of customer service.

CHAPTER : 1 INTRODUCTION

INTRODUCTION

Artificial intelligence (AI) has influenced how we engage in our every day activities by designing and evaluating advanced applications and devices, called intelligent agents, which can perform various functions. A chatbot is ‘‘A computer program designed to simulate conversation with human users, especially over the Internet’’. It uses Natural Language Processing (NLP) and sentiment analysis to communicate in human language by text or oral speech with humans or other chatbots. Apart from imitating human interaction and amusing people, chatbots are useful in various other fields in education, business and ecommerce, health, and entertainment.

1.1 BACKGROUND

- Chatbots has emerged as a hot topic in the latest years, and it is used by numerous companies in various areas - help desk tools, automatic telephone answering systems, e-commerce and so on. Even though the technology has been around since the 60’s (Atwell & Shawar, 2007).
- In the article Chatbots: Are they really useful? Atwell and Shawar provide real-life examples of different chatbots in different contexts. One of the examples is Sophia, a robot that was developed to assist in mathematics at Harvard by answering students questions. This turned out to be applicable in many other contexts.
- Living in Norway you have probably noticed “Kommune Kari”. A chatbot that many of the municipality have available on their web-pages. Kari is there to answer “easy” questions like “when will the garbage truck come?” and “where can I find available jobs?”. Kari’s goal and the job is to provide information so that you as a user don’t have to navigate the “massive information flow”
- (Schibevaag, 2017). This way of using a chatbot is a part of the Question Answering (QA) field which is a combination between AI and information retrieval (Molla & Vicedo, 2007).
- QA can be defined as: “... the task whereby an automated machine (such as a computer) answers arbitrary questions formulated in natural language. QA systems are especially useful in situations in which a user needs to know a very specific piece of information and does not have the time—or just does not want—to read all the available documentation related to the search topic in order to solve the problem at hand”. (Molla & Vicedo, 2007).

1.2 OBJECTIVES

Chatbots boost operational efficiency and bring cost savings to businesses while offering convenience and added services to internal employees and external customers. They allow companies to easily resolve many types of customer queries and issues while reducing the need for human interaction.

Identifying Customer Requirements – Helps in identifying the best products for different customers.
It uses prediction to find the factors that may attract new customers.

Customer Profiling – helps to determine what kind of people buy what kind of products.

1.3 PURPOSE, SCOPE AND APPLICABILITY

Purpose:

There are currently over 70 thousand Alexa skills worldwide. Within this vast array of skills, there are skills for every need. Perhaps the phrase "there's an app for that" needs a little updating.

Some skills have the purpose to inform e.g. BBC News, whilst others have the purpose of entertainment (The Harry Potter Quiz being my personal favourite).

But when should a chatbot inform, and when should it entertain? And more importantly, when should it do both?

To Inform or Instruct

The main incentive to build a chatbot is to inform or instruct. Their purpose is to make our lives easier.

“Tell me what is next on my timetable.”

“What are today’s headlines?”

“Turn the kitchen lights on.”

Talking at a much faster pace than typing, chatbots allow information to be found much more quickly. And in most cases, an entertaining response to “turn the lights on” would be met with frustration, not entertainment, as it would be seen as a barrier to completion.

CHAPTER :2 SURVEY OF TECHNOLOGIES

Nowadays Machine Learning is helping the Retail Industry in many different ways. You can imagine that from forecasting the performance of sales to identify the buyers, there are many applications of machine learning(ML) in the retail industry. “Chatbot Basket Analysis” is one of the best applications of machine learning in the retail industry.

Chatbot Basket Analysis(MBA) also known as association rule learning or affinity analysis, is a data mining technique that can be used in various fields, such as Chatboting, bioinformatics, education field, nuclear science etc. The main aim of MBA in Chatboting is to provide the information to the retailer to understand the purchase behaviour of the buyer, which can help the retailer in correct decision making. There are various algorithms are available for performing MBA. The existing algorithms work on static data and they do not capture changes in data with time. But proposed algorithm not only mine static data but also provides a new way to take into account changes happening in data. The basic method used here is the apriori algorithm, where based on few conditions and criteria the frequent itemset can be discovered and hence be used to the customers needs. This helps in knowing what could be a win-win for both the customer and the buyer. The technique if predominantly used in online shopping. The discovery of these associations can help retailers develop Chatboting strategies by gaining insight into which items are frequently purchased together by customers.

CHAPTER : 3 REQUIREMENT ANALYSIS

3.1 PROBLEM STATEMENT :

Artificial intelligence chatbot is a technology that makes interactions between man and machines using natural language possible. From literature, we found out that in general, chatbot are functions like a typical search engine. Although chatbot just produced only one output instead of multiple outputs/results, the basic process flow is the same where each time an input is entered, the new search will be done. Nothing related to previous output. This research is focused on enabling chatbot to become a search engine that can process the next search with the relation to the previous search output. In chatbot context, this functionality will enhance the capability of chatbot's input processing.They are unaware of the purchasing habits of the customer so they don't know which items should be placed together in their store. With the help of this application shop managers can determine the strong relationships between the items which ultimately helps them to put products that co-occur together close to one another. Also decisions like which item to stock more, cross selling, up selling, store shelf arrangement are determine.

3.2 REQUIREMENT SPECIFICATION :

Our Basket Analysis system itself gathers Frequent Itemset from set of Transaction & other resources, which are then classified according to their semantic orientation and intensity. The proposed system is software which will collect frequent items from multiple transactional databases and then obtained data will be analyzed for association mining. There are multiple algorithms available for association rule mining out of those Apriori Algorithm is used in our Basket Analysis system. In our Basket Analysis system the software will present support and confidence for association rule mining which will use Apriori algorithm by default for frequent itemset mining. Chatbot Basket Analysis takes data at transaction level, which lists all items bought by a customer in a single purchase. The technique determines relationships of what products were purchased with which other product(s). These relationships are then used to build profiles containing If-Then rules of the items purchased. The rules could be written as:

if {A} then {B}.

The if part of the rule (the {A} above) is known as the antecedent and the then part of the rule is known as the consequent (the {B} above). The antecedent is the condition and the consequent is the result. The association rule has three measures that express the degree of confidence in the rule, Support, Confidence, and Lift.

Lift is obtained first because it provides information on whether an association exist or not or if the association is positive or negative. If the value for lift suggests that there is an existence of association rule, then we obtain the value for support.

$\boldsymbol{Lift} = \boldsymbol{P(A \cap B)}\ \boldsymbol{P(A) * P(B)}$

Support of an item or itemset is the fraction of transactions in our dataset that contain that item or itemset. It is an important measure because a rule that have low support may occur simply by chance. A low support rule may also be uninteresting from a business perspective because it may not be profitable to promote items that are seldom bought together. For these reasons, support is often used to eliminate uninteresting rules.

$\boldsymbol{Support} = \boldsymbol{P(A \cap B)/N}$

Confidence is defined as the conditional probability that shows that the transaction containing the LHS will also contain RHS. Association analysis results should be interpreted with caution. The
inference made by an association rules does not necessarily imply causality. Instead, it suggests a strong co-occurrence relationship between the items in the antecedent and consequent of the rule.

$\boldsymbol{Confidence} = \boldsymbol{P}(\boldsymbol{A} \cap \boldsymbol{B})\ \boldsymbol{P}(\boldsymbol{A})$

Confidence and support measure the strength of an association rule. Since the transactional database is quite large, there is a higher risk of getting too many unimportant and rules which may not be of our interest. To avoid these kinds of errors we commonly define a threshold of support and confidence prior to the analysis, so that only useful and interesting rules are generated in our result. If lift is greater than 1, it suggests that the presence of the items on the LHS has increased the probability that the items on the RHS will occur on this transaction. If the lift is below 1, it suggests that the presence of the items on the LHS make the probability that the items on the RHS will be part of the transaction lower. If the lift is 1, it suggests that the presence of items on the LHS and RHS are independent: knowing that the items on the LHS are present makes no difference to the probability that items will occur on the RHS. While performing Chatbot basket analysis, we look for rules with a lift of more than one. It is also preferable to have rules which have high support as this will be applicable to a large number of transactions and rules with higher confidence are ones where the probability of an item appearing on the RHS is high, given the presence of items on the LHS.

Output From the Chatbot Basket Analysis

- **Summary of the Apriori Association Rules.** This is the title of the output. Apriori is the best known algorithm to mine association rules. Apriori iteratively discovers pairs with the largest frequencies and then with decreasing frequencies.

- **Number of Rules: 80.** The number indicates how many rules are generated from the data with the parameters selected.
- **Summary of the Measures of Interestingness.** This is a summary of the descriptive statistics of the distribution values for Support, Confidence, and Lift.
- **Summary of the execution of the apriori commands.**

This is a summary of the settings that come with the apriori algorithm. Except for Support and Confidence, which you can change in the GUI, the remaining settings are set to default values.

3.3 SOFTWARE AND HARDWARE REQUIREMENTS

Hardware Interfaces

Hardware requirements

Chatbot Recommendation

Application server optimum requirements

32 GB RAM

8 Octa Core Processor

500 GB hard disk space1

Ensure C: drive has 100 GB plus free hard disk space.

Database server optimum requirements

Server Recommendation

Chatbot application server

32 GB RAM

8 Core Processor

500 GB hard disk space

For all Chatbot-related database servers

16 GB RAM

8 Core Processor

500 GB hard disk space

Note: If hosting Chatbot databases along with other application databases, ensure the hardware resources are increased proportionately.

Microsoft Azure SQL Database: production environment

vCore model (recommended)

DTU model (Premium tier recommended)

Amazon RDS: production environment db.t3.2xlarge or db.t3.xlarge

Software Interfaces

The following software is required for chat Bot installation:

Software Details

Database Management System See Chatbot database compatibility matrix for a list of compatible versions.

Automation Anywhere Control Room See Chatbot version compatibility matrix for a list of compatible versions.

Supported web browsers

Google Chrome

Microsoft Internet Explorer (Version 11.3.3 onward)

For the prototype we will launch the portal over the internet and other than the hardware specified in the hardware interface section, the software requirements are to support windows operating system with support to MySQL, Apache and PHP servers.

For the data gathering Kaggle is the only source and using Streaming API that offers high throughput. Using this API is perfect because we can retrieve real time information and also this continuous stream will be retrieved with no end and capturing all the messages in the stream without missing any information. The information retrieved in JSON format.

3.4 PRODUCT DOCUMENTATION:

Product descriptive

Chatbots have been used in instant messaging apps and online interactive games for many years and only recently segued into B2C and B2B sales and services.

Organizations can use chatbots in the following ways:

Online shopping. In these environments, sales teams can use chatbots to answer noncomplex product questions or provide helpful information that consumers could search for later, including shipping price and availability.
Customer service. Service departments can also use chatbots to help service agents answer repetitive requests. For example, a service rep might give the chatbot an order number and ask when the order shipped. Generally, a chatbot transfers the call or text to a human service agent once a conversation gets too complex.
Virtual assistants. Chatbots can also act as virtual assistants. Apple, Amazon, Google and Microsoft all have forms of virtual assistants. Apps, such as Apple's Siri and Microsoft's Cortana, or products, like Amazon's Echo with Alexa or Google Home, all play the part of a personal chatbot.

USER DOCUMENTATION

User manual and CD will be made available for troubleshooting and help. Also this will represent as a full backup of the system. The user manual will contain detailed information about the usage of the product from a layman perspective to an advanced network/system administrator. The manual may also be made available online however this manual will be made for the product version but not for the prototype.

TECHNICAL DOCUMENTATION

Technical manual will be made for the purpose of current and future developers involved in the product to understand and follow the solution at the level of coding and the programming languages used.   define the minimum support and confidence for the association rule.Find out all the subsets in the transactions with higher support(sup) than the minimum support. Find all the rules for these subsets with higher confidence than minimum confidence. Sort these association rules in decreasing order. Analyze the rules along with their confidence and support.

PRODUCT CONSTRAINTS

Chatbot basket analysis on it's own will still leave room for improvement. Averages tend to lie. If we are trying to duplicate a conclusion drawn on chain wide data to merchandise a single store, we will hit some speed bumps. It involves analyzing large data sets, such as purchase history, to reveal product groupings, as well as products that are likely to be purchased together.

PROCESSING POWER

Data mining using association rules is also known as "Chatbot-basket analysis." When you visit your local grocery store, you may find that the seafood department has lemons or tartar sauce next to the fish. Apriori is one of the algorithms that is used for frequent pattern mining. In this article, I am going to explain how to apply the "Apriori Algorithm" with Spark in Python.

3.5 CONCEPTUAL MODELS

Since it is a supervised learning task we are provided with a training data set which consists of Tweets labeled with "1" or "0" and a test data set without labels. The training and test data sets can be found here.

label "0": Positive Chatbot analysis

label "1": Negative Chatbot analysis

Now we will read the data with pandas

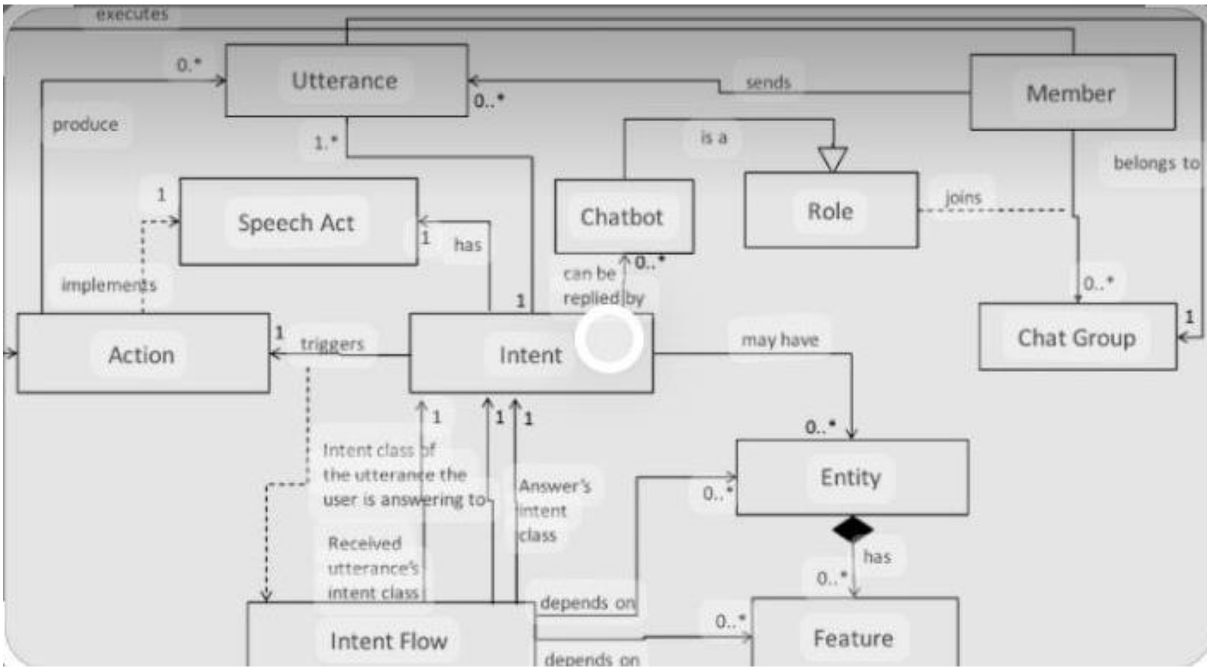
Exploratory Data Analysis

It is necessary to do a data analysis to machine learning problem regardless of the domain. Let's do some analysis to get some insights. The above two graphs tell us that the given data is an imbalanced one with very less amount of "1" labels and the length of the tweet doesn't play a major role in classification. Dealing with imbalanced data is a separate section and we will try to produce an optimal model for the existing data sets.

Data preprocessing and Feature Engineering

The given data sets are comprised of very much unstructured tweets which should be preprocessed to make an NLP model. In this project, we tried out the following techniques of preprocessing the raw data. But the preprocessing techniques is not limited.

- Removal of punctuations.
- Removal of commonly used words (stop words).
- Normalization of words.



In an NLP task the stop words (most common words e.g: is, are, have) do not make sense in learning because they don't have connections with sentiments. So removing them saves the computational power as well as increases the accuracy of the model. All the unusual symbols and the numerical values were removed and returned a pure list with words as shown above. But still we may encounter multiple representations of the same word.(e.g: play, plays, played, playing) Even though the words are different they bring us the same meaning as the normal word "play". So we need to do Lexicon Normalization approach to solve this issue. NLTK's built-in WordNetLemmatizer does this requirement. Now we have done our text preprocessing part and we will move onto the Vectorization and Model Selection.

CHAPTER : 4 SYSTEM DESIGN

4.1 BASIC MODULES

Numpy

NumPy is a module for Python. The name is an acronym for "Numeric Python" or "Numerical Python".NumPy enriches the programming language Python with powerful data structures, implementing multi-dimensional arrays and matrices.

Pandas

Pandas is an open source library in python. It provides ready to use high - performance data structures and data analysis tools. Pandas module runs on top of numpy and it is popularly used for data science and data analytic.

Matplotlib

Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits.

Apriori

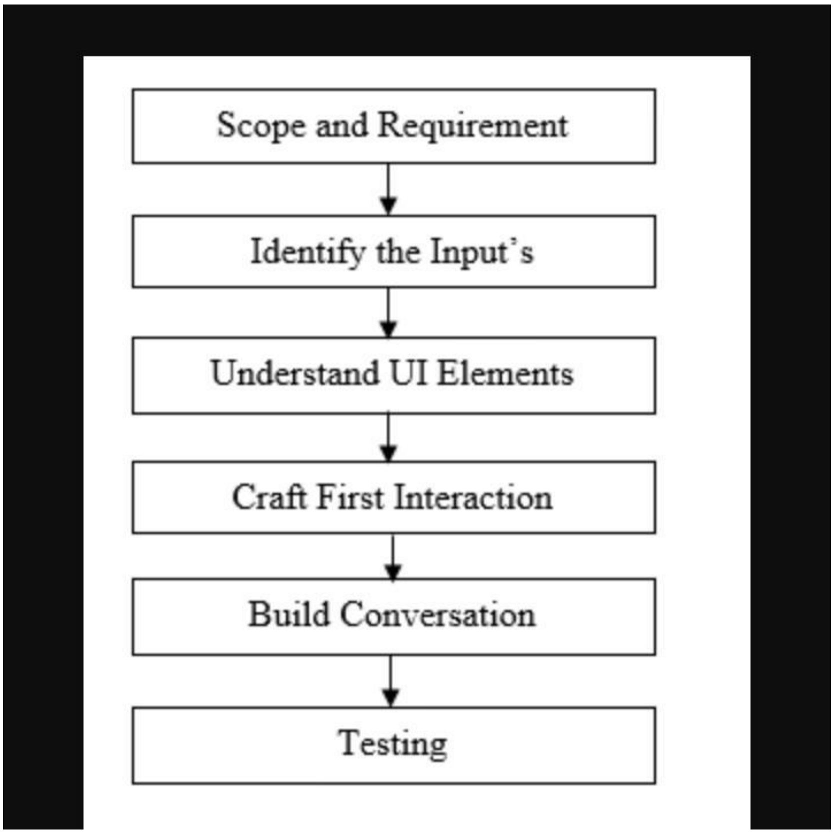
Apriori algorithm is a machine learning model used in Association Rule Learning to identify frequent itemsets from a dataset. This model has been highly applied on transactions datasets by large retailers to determine items that customers frequently buy together with high probability.

4.2 DATA DESIGN

Data is a key component of developing Expert System for Chatbot basket analysis. Getting data safely stored, analyzed and updated in databases is a crucial factor for discovering knowledge and its quality.Data forms the cornerstone of our product development process; it can quickly inform development priorities for enhanced user experience, improved user satisfaction and increased adoption rates. It involves analyzing large data sets, such as purchase history, to reveal product groupings, as well as products that are likely to be purchased together.

4.3 SYSTEM DESIGN:

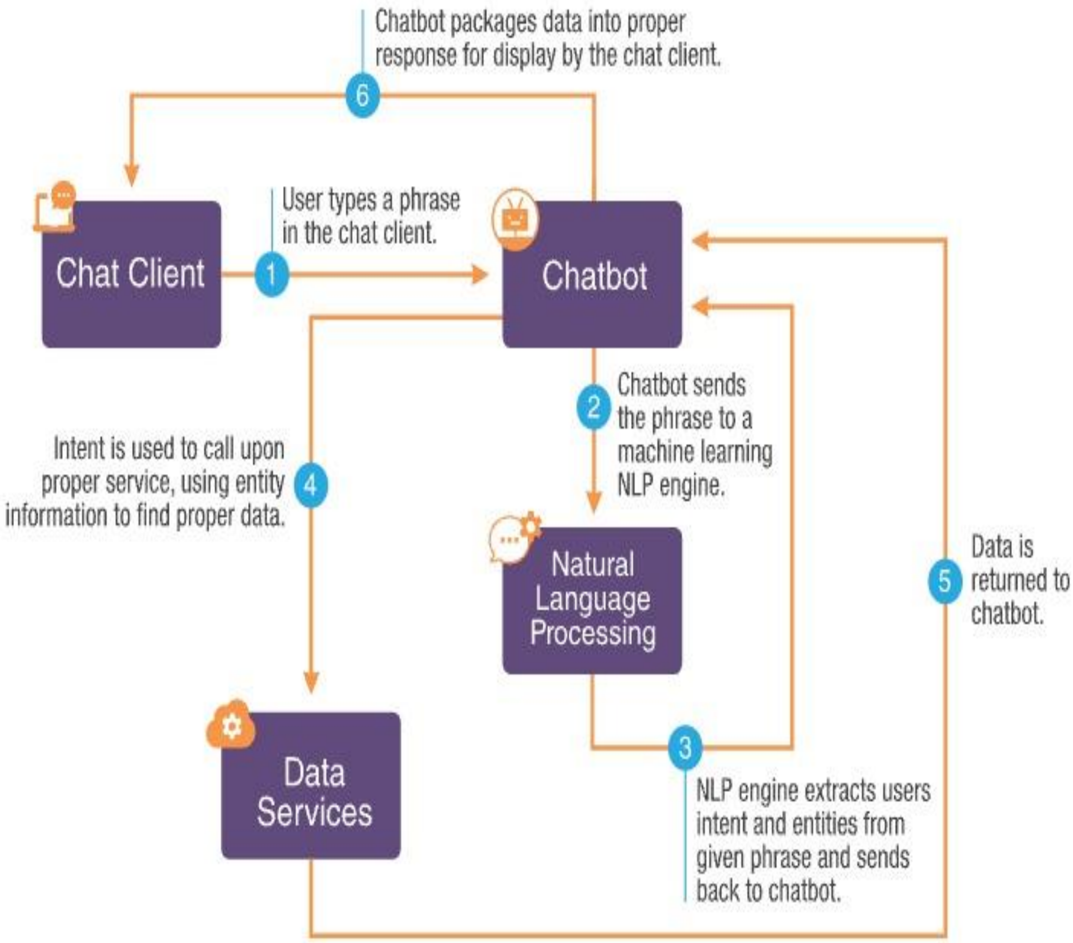
There are seven steps to design the Chatbot process they are scope and requirement, identifying the inputs, understanding the UI elements, craft first interaction, build conversation and finally testing. The Chatbot design process figure is shown in the below



4.4 SCHEMA DESIGN

A schema is a document that lays out the logical structure of a database and translates the data model into specific tables, columns, keys and interrelations. Good database schema design is essential to making your data tractable so that you can make sense of it and build the dashboards and reports you need. A schema organizes data into tables with appropriate attributes, shows the interrelationships between tables and columns, and imposes constraints such as data types. A well-designed schema in a data warehouse makes life easier for analysts by

- removing cleaning and other preprocessing from the analyst’s workflow
- absolving analysts from having to reverse-engineer the underlying data model
- providing analysts with An easily understood starting point for analysis



4.5 APPLICATIONS

Education Environments

The growing demand for learning leads to high competition in higher education institutions. One of the critical reasons for sparse learning and high dropout rates is the fact that, when the number of students grows, the assistance the students get from their teacher is reduced. Chatbots, with their ability to provide educational content and personal assistance, come to support other e-learning practices.

Chatbots for learning support can preserve information by repeating old lessons when the students miss them. They also gather information during a course, which helps the improvement of the learning process and teaching. Students are facilitated in the study as chatbots can answer questions concerning the educational material. A chatbot can also help students with school administration issues, such as enrolling in a course, the exam schedule, their grades, and other related details to their studies so that the pressure on the school departments is considerably reduced.

Customer service

The development of new technologies has made people interact with each other differently, and so has their interaction with businesses. E-commerce has evolved and completely changed the way companies sell their products, but there are some problems related to the quality of customer service. Especially in live chats, the waiting time for a business employee to respond may long, and the answers may not always be relevant

Many companies use chatbots to support customers. Customer care is available 24 h a day via the chatbot, enabling consumers to post their request regardless of the standard operating hours, which enhances user satisfaction. In Gupta et al. (2015), a website based chatbot written in RiveScript helps customers decide which product is suitable for them. Another chatbot implemented with AIML and LSA uses a dataset of Frequently Asked Questions (FAQs) to respond to the users.

Health

In health care, chatbots are designed to provide patients with customized health and therapy information, patient-related products and services, and offer diagnosis and suggest treatments based on patient symptoms.
The advantages of using health care chatbots include encouraging medical decision-making and supporting, improving physical exercise, support of cognitive-behavioral therapy, and somatic disorders that deliver efficient health treatment with precision equal to that of human doctors.

Patients find that chatbots are more reliable contact partners than human physicists; they share more patient knowledge and disclose more symptoms. However, chatbots in healthcare, are generally associated with weak patient adherence because of the perceived lack of consistency or transparency represented by chatbots, as opposed to regular meetings with human doctors. On the other hand, physicians believe that chatbots are more effective in administrative activities such as arranging appointments, finding hospitals, and delivering prescription reminders. Still, they are associated with significant risks, including incorrect medical knowledge. Therefore, physicians do not trust chatbots to replace complicated decision-making tasks that require professional medical advice. Especially in the field of psychiatry, chatbots offer the potential of a new and impactful tool.

Robotics

The most crucial area of research on chatbots is the natural language interface, which is a critical area for physical robots too. Therefore, in the field of physical robots, we find abundant applications of natural language. For example, a novel natural language interface is developed for the autonomous robot called KAMRO. Another natural language interface for instructions to a vision-based robot is designed by (S. (Lauria, Bugmann, Kyriacou, Bos, & Klein, 2001)). Natural language interface allows users to teach vision knowledge and assembly plans to a physical robot. The user can ask questions on vehicle behavior

Industrial use cases

At the current stage of technological evolution, chatbots are already widely used by many companies and organizations. Here are some examples to give the reader an idea of what is going on in practice. 14 E. Adamopoulou and L. Moussiades Machine Learning with Applications 2 (2020) 100006 In the banking sector, chatbots talk to customers and, among other services, provide information about their account balances, facilitate their bill payments, suggest ways to save resources and help activate cards. At the same time, they assist the Bank in collecting feedback from customers. Examples of such chatbots (k, 2020) are Bank Of America's Erika, HDFC's EVA, and Bank of Australia's Ceba.

In the food industry, chatbots accept and track orders, arrange delivery details, make reservations, ask for customer feedback, inform customers on offers and discounts and answer customer questions based on the company's FAQs. Examples of such chatbots are the Pizza-bot of Dominos, , the Messenger bot of Whole Food, , the SubWay's chatbot and the Burger King's chatbot.
IKEA, a leading furniture brand, has launched the so-called ORC chatbot. Zalando, a fashion brand, uses chatbot for order tracking. PVR Cinemas, a large chain of movie theaters in India, uses chatbot for ticket booking. USA's National Railroad Passenger Corporation uses a chatbot called Julie for ticket booking. World Health Organization uses a chatbot called the WHO Health Alert to provide information related to coronavirus.

4.6 ALGORITHMS

There are two algorithms used in the chatbot they are Naïve Bayes Algorithm and Support Vector Machines

Naïve Bayes Algorithm

Naïve Bayes algorithm attempts to classify text into certain categories so that the chatbot can identify the intent of the user, and thereby narrowing down the possible range of responses. Since intent identification is one of the first and foremost steps in chatbot conversations, it is imperative that this algorithm works properly. The algorithm relies on commonality, which essentially means that certain words should have more weight for particular categories based on the frequency of their appearances in that category

Support Vector Machines

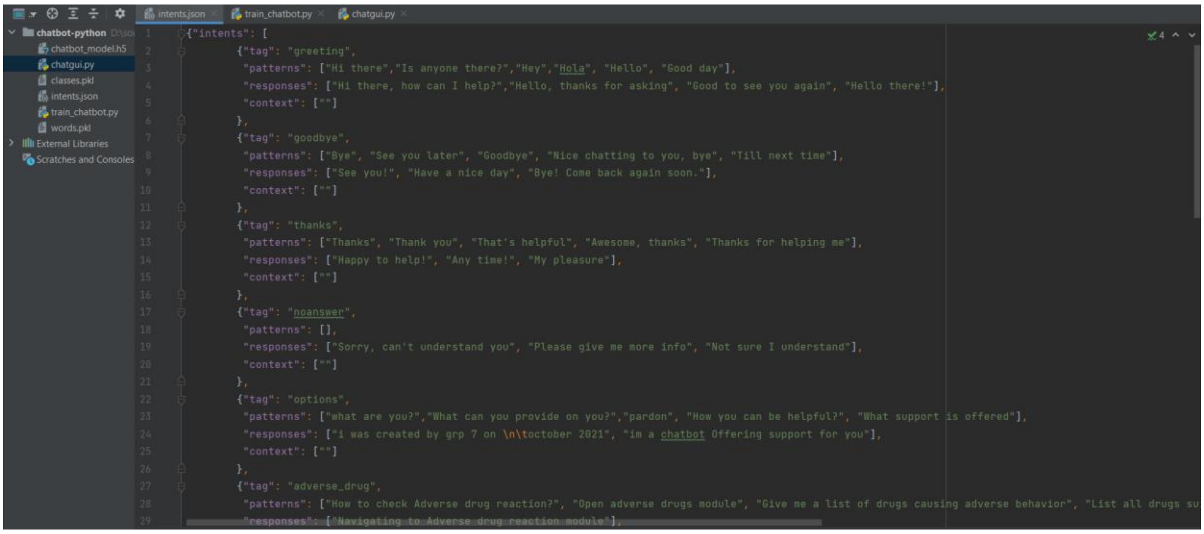
SVMs work based on the principle of Structural Risk Minimization Principle. SVMs work very well with text data and Chatbots because of the high dimensional input space due to large number of text features, linearly separable data and the prominence of sparse matrix. It is one of the most popularly used algorithms for text classification and intent identification.

CHAPTER : 5 IMPLEMENTATION AND TESTING

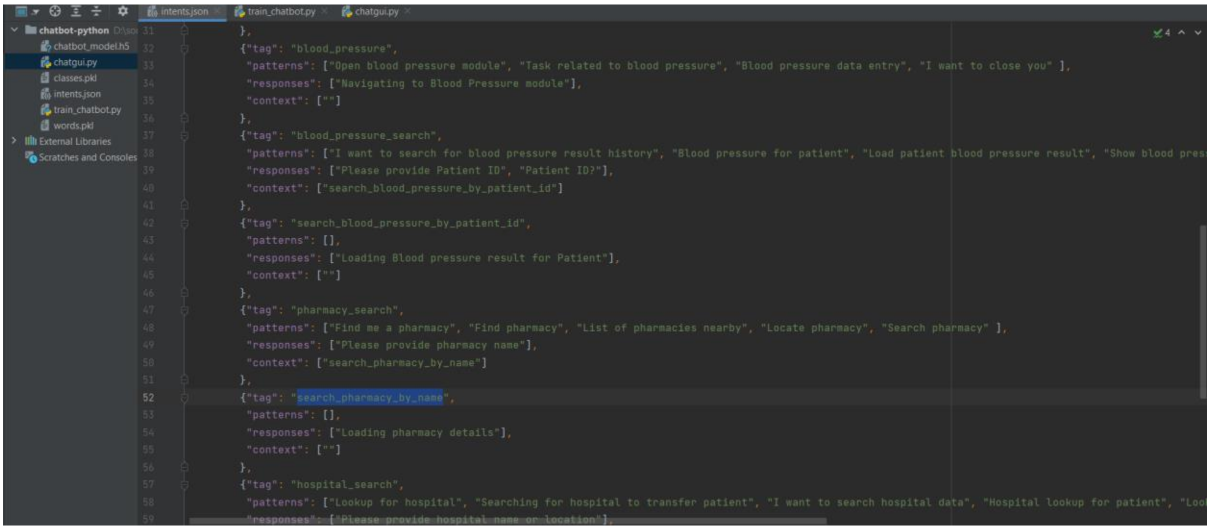
5.1 IMPLEMENTATION APPROACHES

5.2 CODING DETAILS

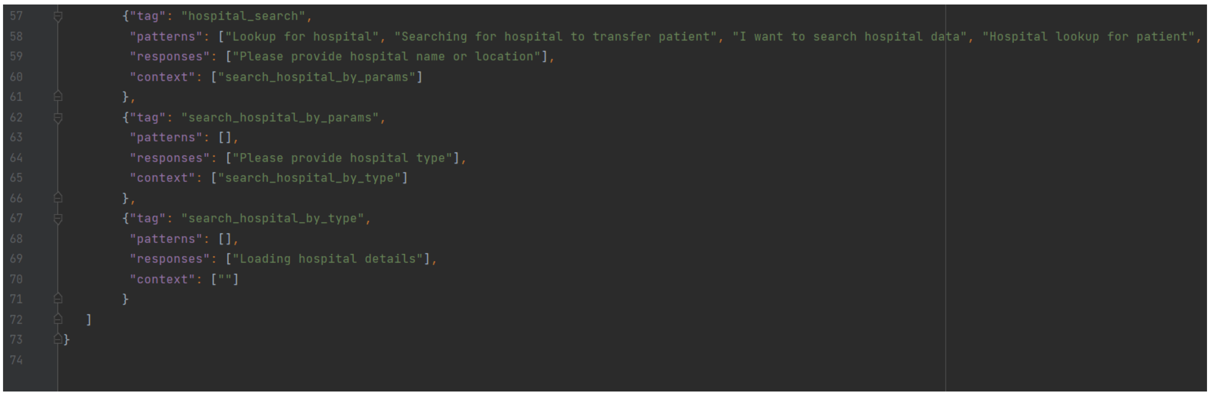
intents. JSON FILE:



```
{"intents": [
        {"tag": "greeting",
         "patterns": ["Hi there","Is anyone there?","Hey","Hola", "Hello", "Good day"],
         "responses": ["Hi there, how can I help?","Hello, thanks for asking", "Good to see you again", "Hello there!"],
         "context": [""]
        },
        {"tag": "goodbye",
         "patterns": ["Bye", "See you later", "Goodbye", "Nice chatting to you, bye", "Till next time"],
         "responses": ["See you!", "Have a nice day", "Bye! Come back again soon."],
         "context": [""]
        },
        {"tag": "thanks",
         "patterns": ["Thanks", "Thank you", "That's helpful", "Awesome, thanks", "Thanks for helping me"],
         "responses": ["Happy to help!", "Any time!", "My pleasure"],
         "context": [""]
        },
        {"tag": "noanswer",
         "patterns": [],
         "responses": ["Sorry, can't understand you", "Please give me more info", "Not sure I understand"],
         "context": [""]
        },
        {"tag": "options",
         "patterns": ["what are you?","What can you provide on you?","pardon", "How you can be helpful?", "What support is offered"],
         "responses": ["i was created by grp 7 on \n\toctober 2021", "im a chatbot Offering support for you"],
         "context": [""]
        },
        {"tag": "adverse_drug",
         "patterns": ["How to check Adverse drug reaction?", "Open adverse drugs module", "Give me a list of drugs causing adverse behavior", "List all drugs su
         "responses": ["Navigating to Adverse drug reaction module"],
```



```
        },
        {"tag": "blood_pressure",
         "patterns": ["Open blood pressure module", "Task related to blood pressure", "Blood pressure data entry", "I want to close you" ],
         "responses": ["Navigating to Blood Pressure module"],
         "context": [""]
        },
        {"tag": "blood_pressure_search",
         "patterns": ["I want to search for blood pressure result history", "Blood pressure for patient", "Load patient blood pressure result", "Show blood pres
         "responses": ["Please provide Patient ID", "Patient ID?"],
         "context": ["search_blood_pressure_by_patient_id"]
        },
        {"tag": "search_blood_pressure_by_patient_id",
         "patterns": [],
         "responses": ["Loading Blood pressure result for Patient"],
         "context": [""]
        },
        {"tag": "pharmacy_search",
         "patterns": ["Find me a pharmacy", "Find pharmacy", "List of pharmacies nearby", "Locate pharmacy", "Search pharmacy" ],
         "responses": ["Please provide pharmacy name"],
         "context": ["search_pharmacy_by_name"]
        },
        {"tag": "search_pharmacy_by_name",
         "patterns": [],
         "responses": ["Loading pharmacy details"],
         "context": [""]
        },
        {"tag": "hospital_search",
         "patterns": ["Lookup for hospital", "Searching for hospital to transfer patient", "I want to search hospital data", "Hospital lookup for patient", "Loo
         "responses": ["Please provide hospital name or location"],
```



```
        {"tag": "hospital_search",
         "patterns": ["Lookup for hospital", "Searching for hospital to transfer patient", "I want to search hospital data", "Hospital lookup for patient",
         "responses": ["Please provide hospital name or location"],
         "context": ["search_hospital_by_params"]
        },
        {"tag": "search_hospital_by_params",
         "patterns": [],
         "responses": ["Please provide hospital type"],
         "context": ["search_hospital_by_type"]
        },
        {"tag": "search_hospital_by_type",
         "patterns": [],
         "responses": ["Loading hospital details"],
         "context": [""]
        }
    ]
}
```

5.3 CODING PART

```
import nltk
from nltk.stem import WordNetLemmatizer
lemmatizer = WordNetLemmatizer()
import json
import pickle

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('intents.json').read()
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'])

# lemmaztize and 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")


# 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")
```

5.3TESTING

```python
import nltk
from nltk.stem import WordNetLemmatizer
lemmatizer = WordNetLemmatizer()
import pickle
import numpy as np

from keras.models import load_model
model = load_model('chatbot_model.h5')
import json
import random
intents = json.loads(open('intents.json').read())
words = pickle.load(open('words.pkl','rb'))
classes = pickle.load(open('classes.pkl','rb'))


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)
```

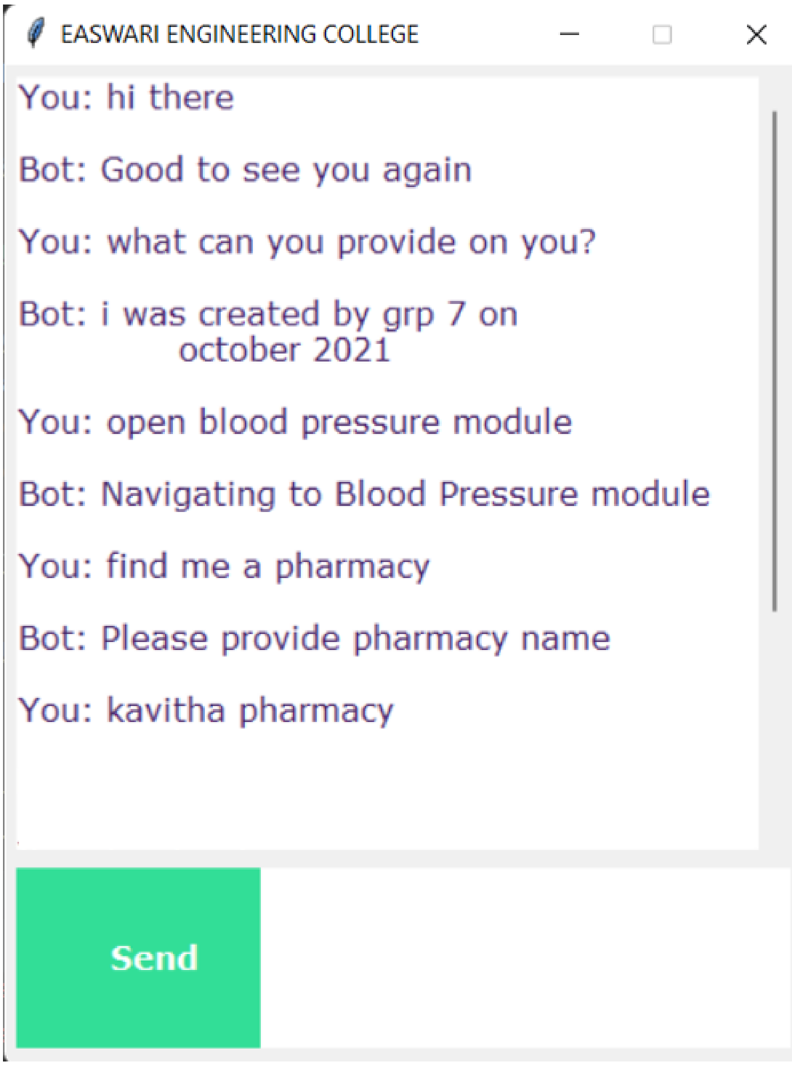
```python
    for s in sentence_words:
        for i,w in enumerate(words):
            if w == s:
                # assign 1 if current word is in the vocabulary position
                bag[i] = 1
                if show_details:
                    print ("found in bag: %s" % w)
    return(np.array(bag))

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
```

```python
            result = random.choice(i['responses'])
            break
    return result

def chatbot_response(msg):
    ints = predict_class(msg, model)
    res = getResponse(ints, intents)
    return res


#Creating GUI with tkinter
import tkinter
from tkinter import *


def send():
    msg = EntryBox.get("1.0",'end-1c').strip()
    EntryBox.delete("0.0",END)

    if msg != '':
        ChatLog.config(state=NORMAL)
        ChatLog.insert(END, "You: " + msg + '\n\n')
        ChatLog.config(foreground="#442265", font=("Verdana", 12 ))

        res = chatbot_response(msg)
        ChatLog.insert(END, "Bot: " + res + '\n\n')

        ChatLog.config(state=DISABLED)
        ChatLog.yview(END)
```

```python

base = Tk()
base.title("EASWARI ENGINEERING COLLEGE")
base.geometry("400x500")
base.resizable(width=FALSE, height=FALSE)

#Create Chat window
ChatLog = Text(base, bd=0, bg="white", height="8", width="50", font="Arial",)

ChatLog.config(state=DISABLED)

#Bind scrollbar to Chat window
scrollbar = Scrollbar(base, command=ChatLog.yview, cursor="heart")
ChatLog['yscrollcommand'] = scrollbar.set

#Create Button to send message
SendButton = Button(base, font=("Verdana",12,'bold'), text="Send", width="12", height=5,
                    bd=0, bg="#32de97", activebackground="#3c9d9b",fg='#ffffff',
                    command= send )

#Create the box to enter message
EntryBox = Text(base, bd=0, bg="white",width="29", height="5", font="Arial")
#EntryBox.bind("<Return>", send)


#Place all components on the screen
scrollbar.place(x=376,y=6, height=386)
ChatLog.place(x=6,y=6, height=386, width=370)
EntryBox.place(x=128, y=401, height=90, width=265)
```

CHAPTER : 6 RESULTS AND CONCLUSION

6.1 RESULTS



6.2 CONCLUSION

When testing the last prototype we got findings suggesting that the participants did not have a problem with getting information from a chatbot instead of a human. The information that they got was not seen as less trustworthy, this could be supported by the fact that the chatbot provided a source for the information it gave. It has been interesting to investigate how the participants interacted with the chatbot and how they reported on it afterwards. Our findings have some indicators leading towards that a chatbot could be a good alternative for acting as a helpful friend for freshmans at a new school. Still we have to stress the fact that the chatbot was not very intelligent and that the evaluators had to adjust their language to match the chatbots.

Because of the scope of the project we did not have time to conduct as much user testing and re-design to the chatbot as we would have liked. This has an impact on the validity of our research. Through the project we have touched on some theory when making the chatbot, but this should also have a larger focus for higher validity. Even though the participants trusted the information given in this project we cannot say that people trusts a chatbot as much as they trust a human being. There are also biases in our project, one of them is that all the students that we included in the project already knew a lot of the answer the prototype could provide. Another bias is that the information the chatbot provides could be seen as “casual” and are not crucial and/or vital This could have had an impact on the results regarding trustworthiness.

With that being said we also think that some of our findings could give some insights into how a very small group of people think about using a chatbot to gain information in a school context. Some of the characteristics of our chatbot was viewed as appropriate for the given context, like “casualness” and links to where the information was gathered. If the IFI chatbot is to be furthered developed, this could be something to draw upon.

6.3 FUTURE SCOPE

Chatbots are fully functioning, semi-autonomous systems that can assist customer service experiences and response time. ... The future scope of chatbots could include many benefits for enterprises, but experts say they will need to be gently nudged in the right direction for businesses to reap these benefits.

Chatbots have an ability to engage customers. They can also foster a relationship between customer and brands, and deliver a more personalized experience. Bots impart information about new product launches and timely updates to the customers.The association rules that we will be getting will display more rapid fluctuation in confidence and lift. It will be interesting to get more insights at a shorter period basis. In future research it will be very interesting to do in-depth understanding of the association rules by evaluating the changes in the lift and confidence values, which can be made possible by calculating the standard deviation.

CHAPTER : 7 REFERENCES

7.1 REFERENCE :

1.M. J. Pereira, L. Coheur, P. Fialho and R. Ribeiro, "Chatbots' greetings to human-computer communication" in arXiv preprint, 2016.

2.O. Deryugina, "Chatterbots", *Scientific and Technical Information Processing*, vol. 37, no. 2, pp. 143-147, 2010.

3.J. S. Malik, P. Goyal and A. K. Sharma, "A comprehensive approach towards data preprocessing techniques & association rules", *Proceedings of The4th National Conference*, 2010.

4.B. A. Shawar and E. Atwell, "Chatbots: are they really useful?", *LDV Forum*, vol. 22, no. 1, pp. 29-49, 2007.

5. B. Setiaji and F. W. Wibowo, "Chatbot using a knowledge in database: Human-to-machine conversation modeling", *Intelligent Systems Modelling and Simulation (ISMS) 2016 7th International Conference on*, pp. 72-77, 2016.

https://www.researchgate.net/publication/351228837_Design_and_Development_of_CHATBOT_A_Review