CHATBOT FOR CUSTOMER SERVICE

END TERM REPORT
By

NAME: Harshpreet Kaur
REG NO. 11802486
ROLL NO. 09

NAME: Shaba Khatun
REG NO. 11802407
ROLL NO. 22

NAME: Shruti Kesharwani
REG NO. 11802458
ROLL NO. 29

NAME: Kankata Manvitha
REG NO. 11807035
ROLL NO. 34

Department of Intelligent Systems
School of Computer Science Engineering
Lovely Professional University, Jalandhar
April 2020

STUDENT DECLARATION

This is to declare that this report has been written by me/us. No part of the report is copied from other sources. All information included from other sources has been duly acknowledged. I/We aver that if any part of the report is found to be copied, I/we are shall take full responsibility for it.

Harshpreet Kaur
09

Shaba Khatun
22

Shruti Kesharwani
29

Kankata Manvitha
34

Place: Lovely Professional University
Phagwara
Punjab

Date: April 2020

TABLE OF CONTENTS

TITLE	PAGE NO.
1. Background and objective of project	5
1.1 Introduction	5
1.2 Objectives	6
2. Description of project	7
2.1 Tokenization	7
2.2 Stemming	7
2.3 Lemmatization	7
2.4 Bag of words	8
3. Work Division	9
3.1 Modules	9
3.2 Distribution	9
4. Implementation of scheduled work	9
5. Output	13
6. Technologies and Framework	13
7. Conclusion	14
8. Future scope	14
9. Github Link	14

BONAFIDE CERTIFICATE

Certified that this project report "CHATBOT for customer service" is the bonafide work of "Harshpreet Kaur, Shaba Khatun, Shruti Kesharwani, and Kankata Manvitha" who carried out the project work under my supervision.

Mr. Dipen Saini
Assistant Professor
Dept. Computer Science and Engineering
Lovely professional University
Phagwara
Punjab

1. Background and objective of project:

1.1 Introduction:

Customer service or customer support is an important point in the user experience, especially on the online scene where most users are extremely demanding both in terms of response time and quality of answers given. Customer support talks to the users regarding their problems and answer accordingly. It is perhaps one of the main aspects of user experience for online services.

A computer program is required to simulate conversation with human users, especially over internet and to provide efficient customer support whenever needed. Chatbots are solution this.

Over last few years, Chatbot's have played a prominent role as human-computer interfaces. Chatbot's are program that attempts to simulate typed conversions, with the aim of making people think they are talking to another person. In scientific literature, chatbots are more formally referred to as conversational agents.
Basically, it is a conversational agent that can interact with user in given subject using the natural language. The underlying principle of every Chatbot is to interact with human user via text messages or even voice messages and behave as though it were capable of understanding the conversation and reply to the user appropriately.

Chatbot's are generally composed of three modules: the user interface, an interpreter, and a knowledge base. Many Chabot's have been deployed on the internet for purpose of education, customer service site, guidance, entertainment etc.

However with the rise of natural language processing, the industry is looking at automated Chatbot solutions to provide quality services to an ever growing user base. Intelligent Chatbot is a project that explores what can be accomplished with AI agents in the world we live in today.

Artificial intelligence has also made significant steps forward in past few years. Techniques such as machine learning and deep learning take advantage of vast amounts of data and cheap processing power to dramatically improve the quality of understanding and decision making.

Messaging applications are the #1 most used applications in the world at the moment and some businesses are already talking advantage of this and developing chatbots of their own to complement their social media presence. With messaging applications taking up the majority of users' time, it makes sense for business to capitalise on this. Bots are becoming smarter and faster and soon will become easier to use than a website or app.

1.2 Objective:

The purpose of this project is to showcase the power of chatbots and how they can be an alternative to using an application or even a website. Chatbots should be easy to use, respond in a timely fashion and be all round user friendly. The bots should make the user interactions as easy and fast as possible to ensure that the users' time is not wasted and that they get what they want without any difficulty or misunderstanding from the bot. The conversation should flow and always keep the user in control of the conversation. Users should come away from their experience with the Chatbot and think that it was fun, easy to use and straightforward interaction that would encourage them to comeback without any hesitation.

For example, a person calling a restaurant to see what time they open at or what is the special today. The customer can simply ask and the bot will respond accordingly.

Our Chatbot provides information about a restaurant named as Pearls inn. It provides information about the timings, signature dish, desserts etc.

2. Description of project:

The chat bot uses various features and packages of python to give users required interface and output.

2.1 Tokenization:

It is very common task in natural language processing. It is basically a task of chopping a character into pieces, called as token, and throwing away the certain characters at the same time, like punctuation. Tokenization is essentially splitting a phrase, sentence, paragraph, or an entire text document into smaller units, such as individual words or terms. Each of these smaller units are called tokens. The meaning of the text could easily be interpreted by analysing the words present in the text. Tokenization can be done as:

- Count number of words
- Count the frequency of word, which is the number of times a particular word is present.

2.2 Stemming:

It is the process of reducing a word to its word stem that affixes to suffixes and prefixes or to the roots of the words known as a lemma. This is proved very useful when we can get the answer by focusing on the main root word rather than focusing on prefix and suffixes which only increase the storage space and are not actually helping to find an optimal solution.
Stemming algorithms are typically rule-based. You can view them as heuristic process

2.3 Lemmatization:

Lemmatization usually refers to doing things properly with the use of a vocabulary and morphological analysis of words, normally aiming to remove inflectional endings only and to return the base or dictionary form of a word, which is known as the lemma. If confronted with the token *saw*, stemming might return just *s*, whereas lemmatization would attempt to return either see or saw depending on whether the use of the token was as a verb or a noun.
lemmatization is a more calculated process. It involves resolving words to their dictionary form.

Text pre-processing includes both Stemming as well as Lemmatization. Many times people find these two terms confusing. Some treat these two as same. Actually, lemmatization is preferred over Stemming because lemmatization does morphological analysis of the words.

2.4 Bag of words:

Bag of words Model is a way of representing text data when modelling text with machine learning algorithm. Bag of Words (BoW) model is a simple algorithm used in Natural Language Processing. In BoW model a sentence or a document is considered as a 'Bag' containing words. It will take into account the words and their frequency of occurrence in the sentence or the document disregarding semantic relationship in the sentences. As machine learning cannot work directly on raw text .The system requires a fixed length inputs and outputs. The system works on numbers, specifically, vectors of numbers. So in natural language processing we use feature extraction. Bag of words involves two things:

- A vocabulary of words known
- A measure of the presence of known words

Steps involved:

- Collect data
- Design the vocabulary
- Create document vectors

After this we get a sparse vector or sparse representation .A more sophisticated approach is to create a vocabulary of grouped words This both changes the scope of the vocabulary and allows the bag-of-words to capture a little bit more meaning from the document.

In above mentioned approach each word or token is called a ‘gram’. Creating a vocabulary of two words is called bigram model.

3. Work division:

3.1 Modules:

Chat (or main function)
Bag of words
Model training (enumerate)
Extracting Information from Json file and performing word tokenization, lemmatization
Making of Json file

3.2 Distribution:

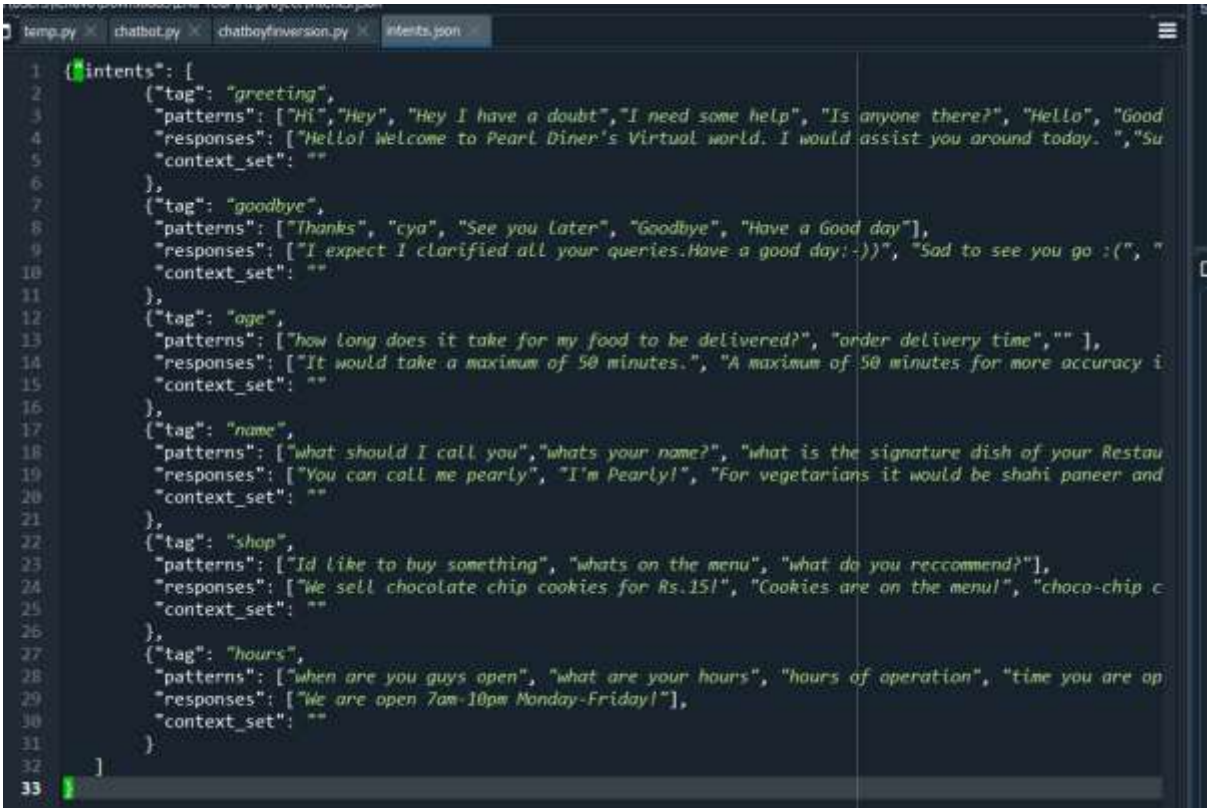
Harshpreet: Extraction information from Json file and performing word tokenization and lemmatization, Report writing
Shaba: Chat (or main function), Making Json file
Shruti: Bag of words, Making Json file
Manvitha: Model training (enumerate), Report writing

4. Implementation of scheduled work:

Shaba:

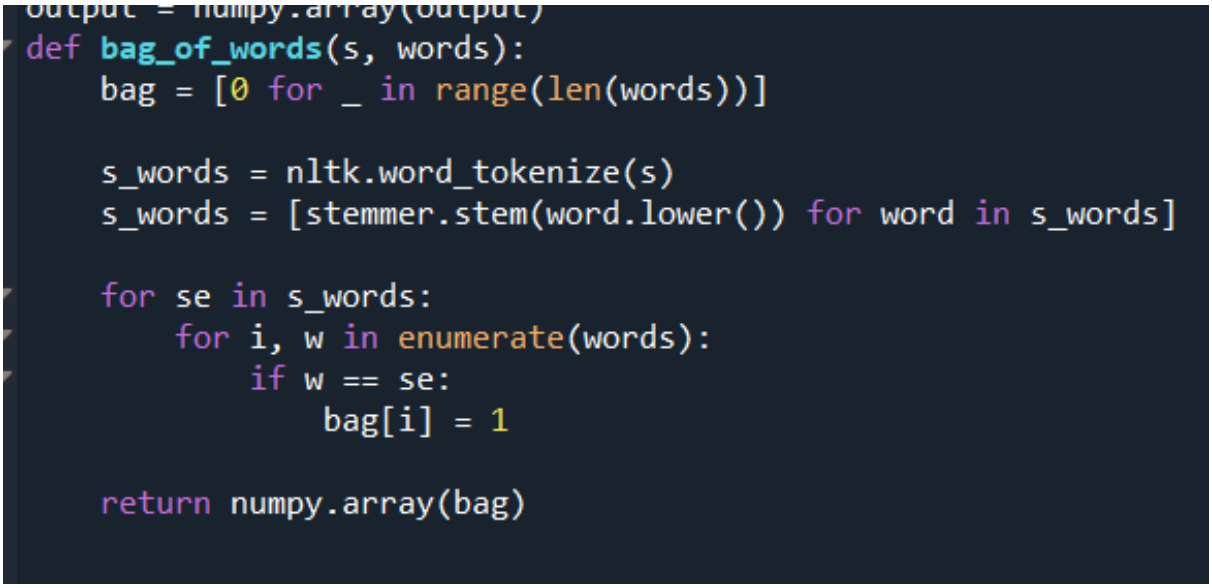
```
85
86  def chat():
87      print("Start talking with the bot (type quit to stop)!")
88      while True:
89          inp = input("You: ")
90          if inp.lower() == "quit":
91              break
92
93          results = model.predict([bag_of_words(inp, words)])
94          results_index = numpy.argmax(results)
95          tag = labels[results_index]
96
97          for tg in data["intents"]:
98              if tg['tag'] == tag:
99                  responses = tg['responses']
100
101         print(random.choice(responses))
102
103 chat()
```

Shaba and Shruti:



```
{"intents": [
        {"tag": "greeting",
         "patterns": ["Hi","Hey", "Hey I have a doubt","I need some help", "Is anyone there?", "Hello", "Good
         "responses": ["Hello! Welcome to Pearl Diner's Virtual world. I would assist you around today. ","Su
         "context_set": ""
        },
        {"tag": "goodbye",
         "patterns": ["Thanks", "cya", "See you later", "Goodbye", "Have a Good day"],
         "responses": ["I expect I clarified all your queries.Have a good day:-))", "Sad to see you go :(", "
         "context_set": ""
        },
        {"tag": "age",
         "patterns": ["how long does it take for my food to be delivered?", "order delivery time","" ],
         "responses": ["It would take a maximum of 50 minutes.", "A maximum of 50 minutes for more accuracy i
         "context_set": ""
        },
        {"tag": "name",
         "patterns": ["what should I call you","whats your name?", "what is the signature dish of your Restau
         "responses": ["You can call me pearly", "I'm Pearly!", "For vegetarians it would be shahi paneer and
         "context_set": ""
        },
        {"tag": "shop",
         "patterns": ["Id like to buy something", "whats on the menu", "what do you reccommend?"],
         "responses": ["We sell chocolate chip cookies for Rs.15!", "Cookies are on the menu!", "choco-chip c
         "context_set": ""
        },
        {"tag": "hours",
         "patterns": ["when are you guys open", "what are your hours", "hours of operation", "time you are op
         "responses": ["We are open 7am-10pm Monday-Friday!"],
         "context_set": ""
        }
    ]
}
```

Shruti:



```
output = numpy.array(output)
def bag_of_words(s, words):
    bag = [0 for _ in range(len(words))]

    s_words = nltk.word_tokenize(s)
    s_words = [stemmer.stem(word.lower()) for word in s_words]

    for se in s_words:
        for i, w in enumerate(words):
            if w == se:
                bag[i] = 1

    return numpy.array(bag)
```

Harshpreet:

```
import json


with open("intents.json") as file:
    data = json.load(file)



words = []
labels = []
docs_x = []
docs_y = []

for intent in data["intents"]:
    for pattern in intent["patterns"]:
        wrds = nltk.word_tokenize(pattern)
        words.extend(wrds)
        docs_x.append(wrds)
        docs_y.append(intent["tag"])

    if intent["tag"] not in labels:
        labels.append(intent["tag"])

words = [stemmer.stem(w.lower()) for w in words if w != "?"]
words = sorted(list(set(words)))

labels = sorted(labels)
```

chatbot.py | chatboyfinversion.py | intents.json

```
for intent in data["intents"]:
    for pattern in intent["patterns"]:
        wrds = nltk.word_tokenize(pattern)
        words.extend(wrds)
        docs_x.append(wrds)
        docs_y.append(intent["tag"])

    if intent["tag"] not in labels:
        labels.append(intent["tag"])

words = [stemmer.stem(w.lower()) for w in words if w != "?"]
words = sorted(list(set(words)))
```

Manvitha:

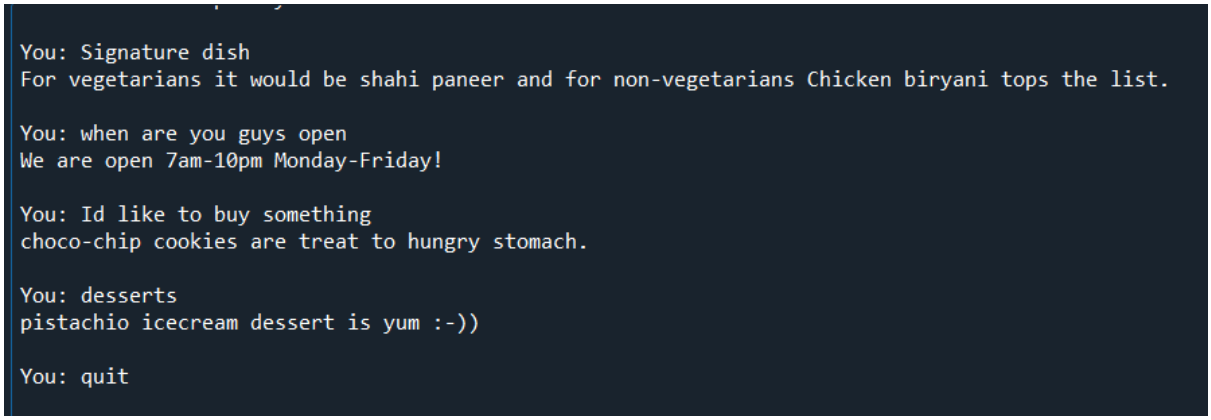
py | chatbot.py | chatboyfinversion.py | intents.json

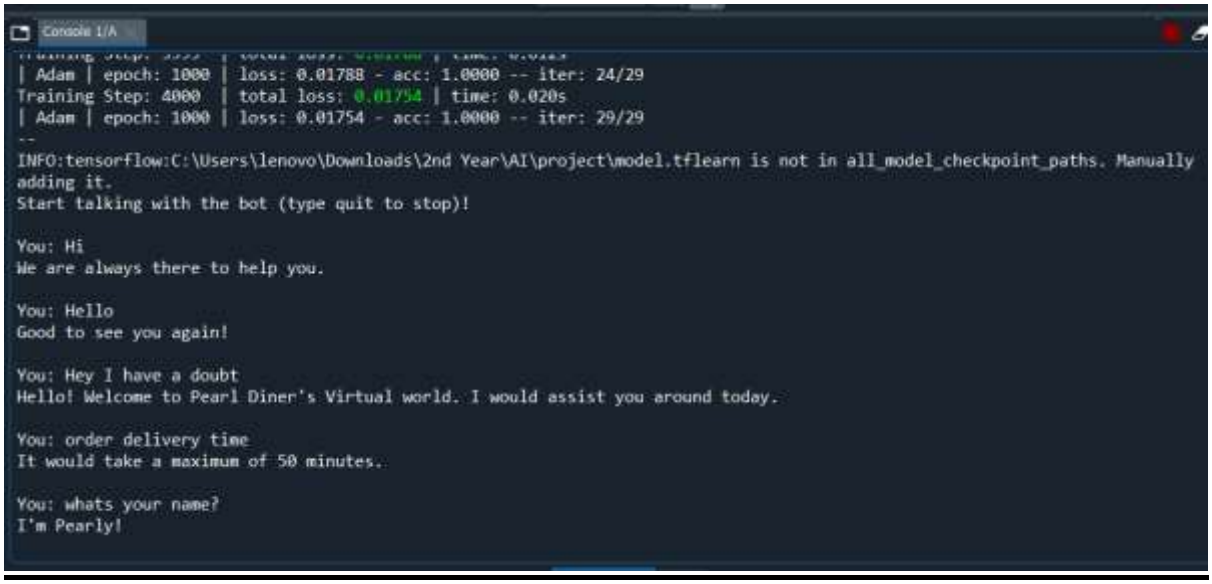
```
for x, doc in enumerate(docs_x):
    bag = []

    wrds = [stemmer.stem(w.lower()) for w in doc]

    for w in words:
        if w in wrds:
            bag.append(1)
        else:
            bag.append(0)

    output_row = out_empty[:]
    output_row[labels.index(docs_y[x])] = 1

    training.append(bag)
    output.append(output_row)
tensorflow.reset_default_graph()

net = tflearn.input_data(shape=[None, len(training[0])])
net = tflearn.fully_connected(net, 8)
net = tflearn.fully_connected(net, 8)
net = tflearn.fully_connected(net, len(output[0]), activation="softmax")
net = tflearn.regression(net)

model = tflearn.DNN(net)
model.fit(training, output, n_epoch=1000, batch_size=8, show_metric=True)
model.save("model.tflearn")

training = numpy.array(training)
output = numpy.array(output)
```

```
labels = sorted(labels)

training = []
output = []

out_empty = [0 for _ in range(len(labels))]

for x, doc in enumerate(docs_x
    bag = []

    wrds = [stemmer.stem(w.lower()) for w in doc]

    for w in words:
```

```
len(o: Sized, /) -> int

Return the number of items in a container.

Click anywhere in this tooltip for additional
```

5. Output:





6. Technologies and frameworks used:

5.1 tenserflow
5.2 tflearn
5.3 pickle
5.4 numpy
5.5 json

7. Conclusion:

A chatbot is an AI agent that can participate in a conversation with a user. Most are equipped with a messenger type interface with an input from a user and an output from the chatbot. The chatbot processes the users input and outputs a reply based on what the user has just sent. It could be a greeting, conversation topic, or even an image.

Chatbots are the new Apps! As we have discussed in the above deliverables, this project brings the power of chatbots and enriches its usability.

Chatbots can give a human like touch to some aspects and make it an enjoying conversation and they are focused entirely on providing information and completing tasks for the humans they interact with. The above mentioned functionality in all the deliverables is implemented and pushed into code.

8. . Future Scope:

The future scope of the project is that as the new information will be updated and added to the existing files or maybe in new files, the Chatbot's ability to answer the queries of the user will become more effective. As one of the major features of Artificial Intelligence is that a system's ability is increased as more and more information and rules are fed to its inferential engine, hence increasing the performance of the system.

9. Github:

- **https://github.com/shabakhatun3012/chatbot**