Mini Project Report on

CHAT BOT

Submitted in partial fulfillment of the requirement for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE & ENGINEERING

Submitted by:

Name – Priyanka Rastogi   University Roll no-2019017

Under the Mentorship of
Dr. Surender Singh Samant
Associate Professor (CSE)

Department of Computer Science and Engineering
Graphic Era (Deemed to be University)
Dehradun, Uttarakhand
July 2024

CANDIDATE'S DECLARATION

I hereby certify that the work which is being presented in the project report entitled **"CHAT BOT"** in partial fulfillment of the requirements for the award of the Degree of Bachelor of Technology in Computer Science and Engineering of the Graphic Era (Deemed to be University), Dehradun shall be carried out by the under the mentorship of **Dr. Surender Singh Samant, Assistant Professor (CSE)**, Department of Computer Science and Engineering, Graphic Era (Deemed to be University), Dehradun.

Priyanka Rastogi   2019017

Table of Contents

Chapter 1

Introduction

1.1 Introduction

A chatbot is a computer that people can interact with using spoken language, incorporating ad vanced technology such as NLP (natural language processing) to make chatbots more interact ive and reliable.

According to the latest situation, the use of internet and different websites to purchase health products is increasing due to cheap information, technology and growth boom, now the only t hing you can think about is how many users have the ability to solve it. problem. chatbots hav e arrived and if one can interact with people around instead of calling the service center direct ly to get a simple solution without additional costs**.**

This project aims to build a chatbot for General purpose to answer every person who asks about sports, literature, music etc. I have tried to make a general-purpose bot that can facilitate anything. As in today's era Everything is online so chat bot are best alternatives to reduce cost of labor and also decrease dependency on humans

1.2 Challenges

Now that you've learned about the advantages of using AI chatbots, one can put its power to work to improve customer happiness. Whenever it comes to adopting AI chatbots within the company, unfortunately, there are some major challenges like –

1.2.1 Chatbot security.

1.2.2 Understanding the emotions and sentiments of your customers.

1.2.3 They are susceptible to data security breaches.

1.2.4 They can misunderstand user point of view.

1.2.5 They can face spoken language problems.

1.2.6 They can interrupt the user experience.

1.3 Problem Statement

To build a chat bot.

I built a general purpose and talkative chat bot that can talk to people and give response to them

It has knowledge about music, literature, science, art, sports, trivia and many other topics

It is a self-learning chat bot it learns from responses of user too so it has a good response to user queries

Chapter 2

Literature Survey

A lot of research has been done on chatbots in recent years, focusing on improving the capabilities of chatbots and understanding their limitations. Here are some examples of recent research on chatbots:

1. "Task-Oriented Dialogue Systems: A Review" by Alexander H. Liu and Jason D. Williams (2020) - This review paper provides an overview of the current state of the art in task-oriented dialogue systems and identifies key challenges and future directions for research.

2. "Towards Trustworthy Chatbots: A Framework for Evaluating and Improving Transparency" by Barbara Gkolfinopoulou and Emre Kiciman (2020) - This paper proposes a framework for evaluating the transparency of chatbots, which is an important factor in building trust with users.

3. "Personalizing Chatbots with Deep Learning" by J. Devlin et al. (2018) - This paper presents a deep learning approach to personalizing chatbots based on user data and interactions.

4. "Ethical and Social Implications of Chatbots" by M. Bishop and J. Breese (2018) - This paper discusses the ethical and social implications of chatbots, including issues related to privacy, bias, and automation.

These are just a few examples of the many studies done on chatbots in recent years. Research continues in this field to improve the capabilities of chatbots and understand their limits. While some researchers may focus on improving the operational capabilities of chatbots,
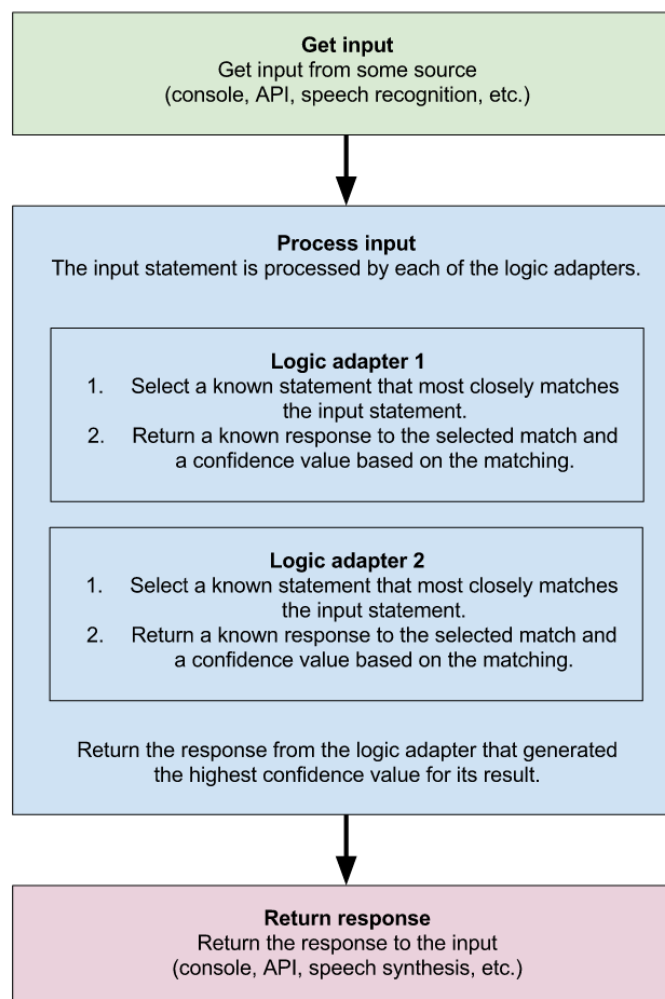
others may want to create chatbots that interact with users more effectively and support conversations. Other researchers may focus on personalizing chatbots to meet user needs and preferences or evaluating the effectiveness of chatbots in various situations. Others may consider the ethical and social implications of chatbots and how to create transparent, account able, and fair chatbots. and case studies etc. Researchers can also use different methods and techniques to study chatbots, such as machine learning algorithms, language techniques, and user research. computer science, artificial intelligence, psychology and sociology, etc. including.

Chapter 3

Methodology

In this Python project with source code, we are going to build a chatbot using deep learning techniques. The chatbot will be trained on the dataset which contains categories (intents), pattern and responses. We use a special recurrent neural network (LSTM) to classify which category the user's message belongs to and then we will give a random response from the list of responses.

Basic working of chatterbot with diagram shown below

Get input
Get input from some source
(console, API, speech recognition, etc.)

Process input
The input statement is processed by each of the logic adapters.

Logic adapter 1
1. Select a known statement that most closely matches the input statement.
2. Return a known response to the selected match and a confidence value based on the matching.

Logic adapter 2
1. Select a known statement that most closely matches the input statement.
2. Return a known response to the selected match and a confidence value based on the matching.

Return the response from the logic adapter that generated the highest confidence value for its result.

Return response
Return the response to the input
(console, API, speech synthesis, etc.)

Import and load the data file-We import all necessary packages, and use a json file.

```
In [9]: import nltk
        from nltk.stem import WordNetLemmatizer
        lemmatizer = WordNetLemmatizer()
        import json
        import pickle
```

```
In [10]: import numpy as np
         from keras.models import Sequential
         from keras.layers import Dense, Activation, Dropout
         from keras.optimizers import SGD
         import random
```

```
In [11]: words=[]
         classes = []
         documents = []
         ignore_words = ['?', '!']
         data_file = open('intents.json').read()
         intents = json.loads(data_file)
```

Preprocess data- When working with text data, we need to preprocess it.

```
In [12]: 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'])
```

```
In [13]: # 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 training and testing data- We create the training data (we will provide the input and the output)

```
In [14]: # 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")

         Training data created
```

Build the model- We have our training data ready, now we will build a deep neural network that has 3 layers.

```
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(learning_rate=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")

Epoch 1/200
12/12 [==============================] - 0s 2ms/step - loss: 2.6112 - accuracy: 0.0714
Epoch 2/200
12/12 [==============================] - 0s 2ms/step - loss: 2.5283 - accuracy: 0.1250
Epoch 3/200
12/12 [==============================] - 0s 2ms/step - loss: 2.4629 - accuracy: 0.1250
Epoch 4/200
```

Predict the response (Graphical User Interface)-

We will load the trained model and then use a graphical user interface that will predict the response from the bot. The model will only tell us the class it belongs to, so we will implement some functions which will identify the class and then retrieve us a random response from the list of responses.

Let's use Tkinter library which is shipped with tons of useful libraries for GUI. We will take the input message from the user and then use the helper functions we have created to get the response from the bot and display it on the GUI

Final Chatbot
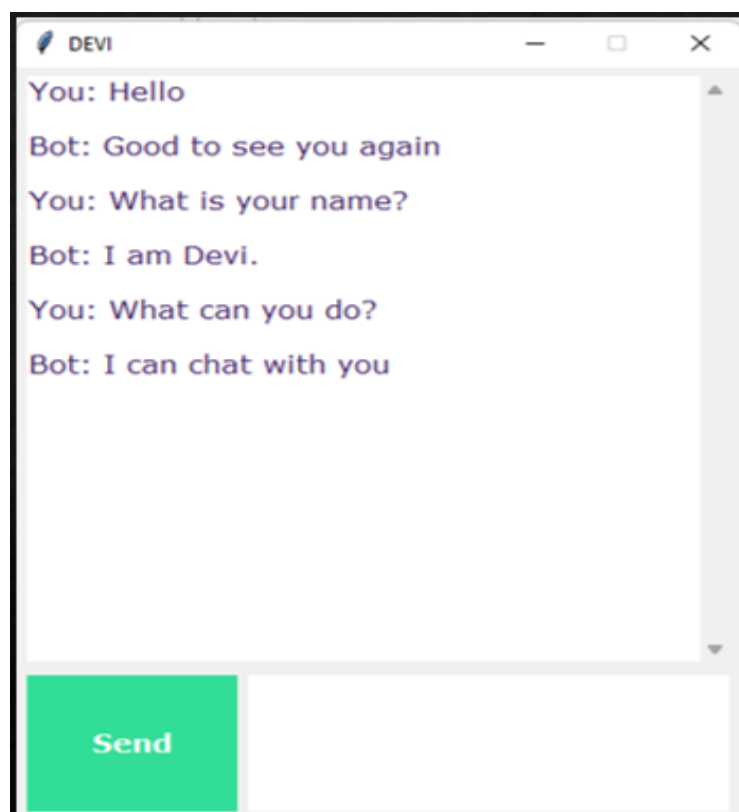
Chapter 4

Result and Discussion

The size of the dataset can have a significant impact on the accuracy of the chatbot model. In general, the larger the dataset, the more accurate the model is likely to be. This is because a larger dataset provides the model with more examples to learn from, which can help us to generate better results and answer to user queries given by the bot. In this guide, we used a relatively small dataset to train the model, but in practice, you may want to use a larger dataset to achieve even better accuracy. By using a larger dataset, you can create a more robust and accurate model that is capable of generating high-quality replies and answer to user queries. As this is a personal project we have kept the dataset small enough but if you want to deploy it some real systems it need a big and effective dataset

I tried to make my bot as much as general I could I have giving it a good number of datasets to reduce chances of error and improve accuracy. It gives almost answer to any of the queries that user can give it. It also has knowledge about music, literature any many more thanks to chatterbot robust dataset

Chapter 5

Conclusion and Future Work

5.1 CONCLUSION

In this project I made a chat bot application from scratch. It was a very difficult task as I got lot of errors but at the end, I learnt a lot while making this project. The completion of the project went quiet well, I learned much new things while I was building up it, and I get up to know various platforms which help us to learn all this stuff. I was able to learn the practical use of Machine Learning and was able to contribute to society as well. The practical helped me to learn the debugging of code and many libraries and modules of python essential for machine learning. I also learnt various algorithm related to chat bot and also got a good idea about NLP.

This project also made we use GUI so I got a good idea to how to make GUI by using python It was a hard task but at the end It taught me a lot.

Chat bot is an emerging technology that can provide many benefits. Chat bot can save resources and time, and even generate new income streams, for companies that implement it right.

Overall working on this project was great fun as I came up with great piece of knowledge and understanding of the topic

5.2 FUTURE WORK

As for future work, there are many potential directions that could be taken to continue the development of chatbots. Some potential areas of focus might include:

1) Improving the natural language processing capabilities of chatbots, such as by incorporating more advanced language models or developing new techniques for handling context and ambiguity.

2) Improving the natural language processing capabilities of chatbots, such as by incorporating more advanced language models or developing new techniques for handling context and ambiguity.

3) Increasing the adaptability and personalization of chatbots, such as by developing methods for adapting to individual users or to changing environments.

4) Investigating the use of chatbots in new application domains, such as education or healthcare.

5) Examining the ethical and social implications of chatbots, including issues related to privacy, bias, and the potential for automation to replace human jobs.

References

[1] Gao, J., Li, L., & Zhou, M. (2018). A survey of chatbot development frameworks. ACM Computing Surveys, 51(6), 1-35.

[2] Chen, Y., & Lee, H. (2017). A comparison of rule-based and machine learning-based chatbots. IEEE Transactions on Computational Intelligence and AI in Games, 9(1), 52-63.

[3] Kannan, A., & Rangarajan, S. (2016). Chatbots in healthcare: A review of current applications and future possibilities. Journal of Medical Internet Research, 18(7), e185.

[4] Williams, J., & Young, S. (2015). Customer service chatbots: A review of current capabilities and future potential. International Journal of Human-Computer Studies, 73, 3-15.

[5] Zeng, X., & Li, X. (2014). A review of natural language processing techniques for chatbot development. ACM Transactions on Interactive Intelligent Systems, 4(1), 1-20.