VELAMMAL
INSTITUTE OF TECHNOLOGY

Approved by AICTE - New Delhi
Affiliated to Anna University - Chennai
Accredited by NBA & NAAC

DEPARTMENTS OF ELECTRONICS AND COMMUNICATION

TEAM MEMBERS:

BOLLEDDU KISHORE (113321106012)

CHILAKA SARATH(113321106015)

KONDURU ABHILASH(113321106046)

MIKILI MANOAJITH(113321106051)

NEERUKATTU MADHAN KUMAR(113321106061)

DEVELOPMENT PART -2
INTRODUCTION

Natural Language processing is one of the advanced fields of artificial intelligence which makes the systems understand and process the human language. The main use-case of NLP can be seen in chatbot development, spam classification, and text summarization. In today's article, we're going to discuss the chatbot application of NLP.

Chatbots are computer software programs that can interact with humans. With the advancement in machine learning mainly natural language processing, everyone started to create intelligent chatbot systems. You can see different types of chatbots on different websites, chatbots for booking airline tickets on Airline company websites, customer support chatbots in different apps, etc... are such examples. Do you want to create one such chatbot!!. Let's have an amazing session on chatbot development in today's article.

Now we can create our virtual environment named my_env, so take the terminal in the vscode or any code editor and write the below code.

```
virtualenv my_env
```

Next is to activate our virtual environment

Activation in windows power shell

```
my_envScriptsactivate.ps1
```

Activation in command prompt

```
my_envScriptsactivate.bat
```

The virtual environment is activated.

Installation of Libraries

Now we have to install the libraries required for this project separately in this environment.

```
pip install keras nltk tensorflow
```

CREATING INTENTS FILE

First of all, let's look into our intents_file.json file. This intents file contains the different patterns of the question that the user might enquire and the possible output for the specific question and a tag for that type of question

```
{"intents": [
    {"tag": "greetings",
     "patterns": ["Hello there", "Hey, How are you", "Hey", "Hi", "Hello", "Anybody", "Hey there'
     "responses": ["Hello, I'm your helping bot", "Hey it's good to see you", "Hi there, how can
     "context": [""]
    },
    {"tag": "thanks",
     "patterns": ["Thanks for your quick response", "Thank you for providing the valuable inform
     "responses": ["Happy to help you", "Thanks for reaching out to me", "it's My pleasure to he
     "context": [""]
    },
    {"tag": "no_answer",
     "patterns": [],
     "responses": ["Sorry, Could you repeat again", "provide me more info", "can't understand yo
     "context": [""]
    },
    {"tag": "support",
     "patterns": ["What help you can do?", "What are the helps you provide?", "How you could hel
     "responses": [ "ticket booking for airline", "I can help you to book flight tickets easily"
     "context": [""]
    },
    {"tag": "goodbye",
        "patterns": ["bye bye", "Nice to chat with you", "Bye", "See you later buddy", "Goodbye"
        "responses": [ "bye bye, thanks for reaching", "Have a nice day there", "See you later"]
        "context": [""]
    }
```

IMPLEMENTATION

Importing some of the required libraries for our project.

```
import numpy as np
import nltk
import json
import pickle
import re
import random
from keras.models import Sequential
from keras.layers import Dense, Activation, Dropout
from tensorflow.keras.optimizers import SGD
from nltk.stem import WordNetLemmatizer
```

PREPROCESSING

Loading the JSON file and reading it. Also, we are initializing some lists for saving the preprocessed and preprocessing data.

```
tokenized_words=[]
classes = []
doc = []
ignoring_words = ['?', '!']
data_file = open('intents_file.json').read()
intents = json.loads(data_file)
```

We loaded the JSON file. Now we have to perform some preprocessing, we are going to iterate through each of the patterns questions in the intents file and tokenize it. This tokenized text along with the tag is stored as a list. tokenized_words contains all the different words in the intents file which is tokenized using nltk.

```
for intent in intents['intents']:
    for pattern in intent['patterns']:
        w = nltk.word_tokenize(pattern) #tokenizing
        tokenized_words.extend(w)
        doc.append((w, intent['tag']))
        if intent['tag'] not in classes:
            classes.append(intent['tag'])
```

Now we have to perform lemmatization on the data and need to remove the question tag and other ignoring words

```
lemmatizer = WordNetLemmatizer()
lemmatized_words = [lemmatizer.lemmatize(words.lower()) for words in tokenized_words if w not in
```

As the next step, we need to create our training data. The input feature is the bag of words model of questions that the user is asking and the output feature is the tag or class that the input question pattern belongs to.

```
training_data = []

empty_array = [0] * len(classes)

for d in doc:

    bag_of_words = []

    pattern = d[0]

    pattern = [lemmatizer.lemmatize(word.lower()) for word in pattern]

    for w in lemmatized_words:

        bag_of_words.append(1) if w in pattern else bag_of_words.append(0)

    output_row = list(empty_array)

    output_row[classes.index(d[1])] = 1

    training_data.append([bag_of_words, output_row])

random.shuffle(training_data)

training = np.array(training_data)

train_x = list(training[:,0])
```

TESTING THE MODEL

Now let's take another python file for testing and creating our actual chatbot

Importing the required libraries.

```
import pickle
import numpy as np
import json
from keras.models import load_model
import random
from nltk.stem import WordNetLemmatizer
lemmatizer = WordNetLemmatizer()
```

Next is to load our models and pickle files that we are saved during the training timing

```
intents_file = json.loads(open('intents.json').read())
lem_words = pickle.load(open('lem_words.pkl','rb'))
classes = pickle.load(open('classes.pkl','rb'))
bot_model = load_model('chatbot_model.h5')
```

Creating a function that takes the user input as a parameter for performing some preprocessing techniques like tokenization and stemming.

```
def cleaning(text):
    words = nltk.word_tokenize(text)
    words = [lemmatizer.lemmatize(word.lower()) for word in words]
    return words
```

Our model requires numerical features for the prediction of classes, so we are creating another function for creating the bag of words model for the preprocessed text.

```
def bag_ow(text, words, show_details=True):
    sentence_words = cleaning(text)
    bag_of_words = [0]*len(words)
    for s in sentence_words:
        for i,w in enumerate(words):
            if w == s:
                bag_of_words[i] = 1
    return (np.array(bag_of_words))
```

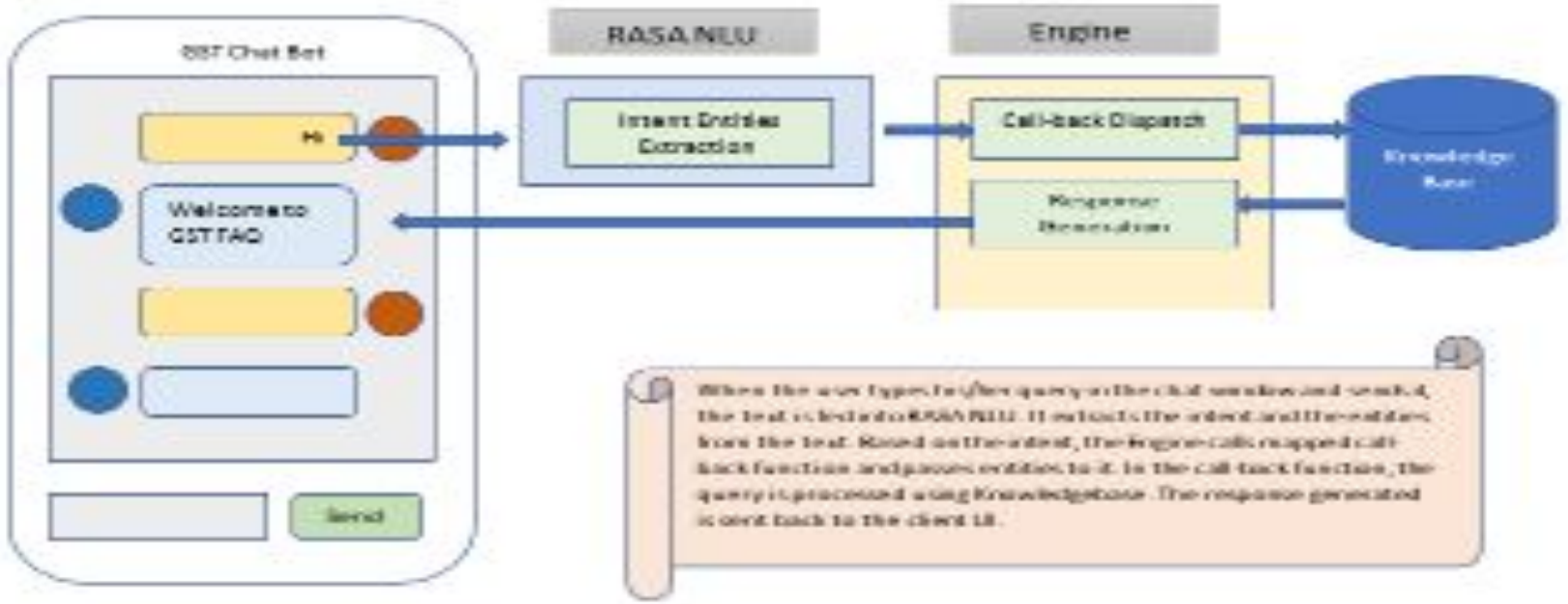
INTERACTING WITH CHATBOT

```
for i in range(3):
  text = input("You : ")
  print("Bot : ",bot_response(text))
```

Output

```
You : hey
Bot :  Hi there, how can I help you?
You : what help can you do
Bot :  ticket booking for airline
You : bye
Bot :  See you later
```

DATA BASE ANALYSIS



CONCLUSION

In this article, we've briefly discussed chatbot development from scratch. And you got the idea about the working flow or data flow of chatbot making and prediction of the response. The main key insights from this article are

- Created a virtual environment for the project
- Created a JSON file including the possible pattern of questions and possible output response
- Preprocessing the text file and the creation of a deep learning chatbot model
- Created several functions for communicating with the chatbot and for receiving the chatbot response

These are the primary outcomes of the above article. You can customize the chatbot by editing the intents JSON file. Try to create your own chatbot by referring to this article. Hope you all liked this article.