Bahria University,
Karachi Campus

LAB EXPERIMENT NO.
11

LIST OF TASKS

TASK NO	OBJECTIVE
1.	Develop a chatbot using Python and NLTK that can handle and respond to user queries by understanding their context, even if the queries do not exactly match the entries in the chatbot's knowledge base.

Submitted On:
6/7/2024

TASK NO 1: Develop a chatbot using Python and NLTK that can handle and respond to user queries by understanding their context, even if the queries do not exactly match the entries in the chatbot's knowledge base.

```
import io
import random
import string
import warnings
import numpy as np
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity
import warnings
warnings.filterwarnings('ignore')
import nltk
from nltk.stem import WordNetLemmatizer
nltk.download('popular', quiet=True)
f=open('/content/chatbot.txt','r',errors = 'ignore')
raw=f.read()
raw = raw.lower()# converts to lowercase
sent_tokens = nltk.sent_tokenize(raw)# converts to list of sentences
word_tokens = nltk.word_tokenize(raw)# converts to list of words
lemmer = nltk.stem.WordNetLemmatizer()
#WordNet is a semantically-oriented dictionary of English included in NLTK.
def LemTokens(tokens):
    return [lemmer.lemmatize(token) for token in tokens]
remove_punct_dict = dict((ord(punct), None) for punct in string.punctuation)
def LbyemNormalize(text):
    return LemTokens(nltk.word_tokenize(text.lower().translate(remove_punct_dict)))
GREETING_INPUTS = ("hello", "hi", "greetings", "sup", "what's up","hey","yo","How are you")
GREETING_RESPONSES = ["hi", "hey", "*nods*", "hi there", "hello", "I am glad! You are talking to me"]
def greeting(sentence):
    for word in sentence.split():
        if word.lower() in GREETING_INPUTS:
            return random.choice(GREETING_RESPONSES)
def response(user_response):
    spar_response=''
    sent_tokens.append(user_response)
    # Use LbyemNormalize, the function you defined
    TfidfVec = TfidfVectorizer(tokenizer=LbyemNormalize, stop_words='english')
    tfidf = TfidfVec.fit_transform(sent_tokens)
    vals = cosine_similarity(tfidf[-1], tfidf)
    idx=vals.argsort()[0][-2]
    flat = vals.flatten()
    flat.sort()
    req_tfidf = flat[-2]
    if(req_tfidf==0):
        spar_response=spar_response+"I don't understand you"
        return spar_response
    else:
        spar_response = spar_response+sent_tokens[idx]
        return spar_response
flag=True
print("Spar: My name is Spar. I will answer your queries about Chatbots. If you want to exit, type Bye!")
while(flag==True):
    user_response = input("You: ")
    user_response=user_response.lower()
    if(user_response!='exit'):
        if(user_response=='thanks' or user_response=='thank you' ):
            flag=False
            print("Spar: You are welcome..")
        else:
            if(greeting(user_response)!=None):
                print("Spar: "+greeting(user_response))
            else:
                print("Spar: ",end="")
                print(response(user_response))
                sent_tokens.remove(user_response)
    else:
        flag=False
```

OUTPUT:

```
Spar: My name is Spar. I will answer your queries about Chatbots. If you want to exit, type Bye!
You: hello
Spar: hey
You: Development
Spar: chatbot development platforms
the process of building, testing and deploying chatbots can be done on cloud based chatbot developmen
You: AI
Spar: one pertinent field of ai research is natural language processing.
You: chatbot
Spar: design
the chatbot design is the process that defines the interaction between the user and the chatbot.the c
You: exit
```