

PHASE-4

CREATE THE CHATBOT USING PYTHON

DEVELOPMENT PART (2) :

Building the chatbot by integrating it into a web app using Flask.

What is Flask tool used for?

Flask is used for developing web applications using python, implemented on Werkzeug and Jinja2. Advantages of using Flask framework are: There is a built-in development server and a fast debugger provided.

Why use Flask for Python?

It offers a straightforward and adaptable method for developing Python-based web applications and APIs (Application Programming Interfaces). Flask is renowned for its straightforward design, which gives developers the freedom to select the elements they desire and customise their apps to meet their needs.

```
{1}: import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
# Input data files are available in the read-only "../input/" directory
# For example, running this (by clicking run or pressing Shift+Enter) will list_
↳ all files under the input directory
import os
for dirname, _, filenames in os.walk('dialogs.txt'):
for filename in filenames:
print(os.path.join(dirname, filename))
```

```
[2]: import numpy as np
import random
import string
import nltk
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity
[3]: f = open("dialogs.txt", "r", errors = 'ignore')
raw_doc = f.read()
raw_doc = raw_doc.lower()
[4]: nltk.download('punkt')
```

```
nltk.download('wordnet')
[nltk_data] Downloading package punkt to
[nltk_data] C:\Users\rohig\AppData\Roaming\nltk_data...
[nltk_data] Package punkt is already up-to-date!
[nltk_data] Downloading package wordnet to
[nltk_data] C:\Users\rohig\AppData\Roaming\nltk_data...
[nltk_data] Package wordnet is already up-to-date!
```

```
[4]: True
```

```
[5]: sent_token = nltk.sent_tokenize(raw_doc)
word_token = nltk.word_tokenize(raw_doc)
print(f"Number of sentence : {len(sent_token)}")
print(f"Number of Words in : {len(word_token)}")
Number of sentence : 8272
Number of Words in : 60702
[6]: lemmer = nltk.stem.WordNetLemmatizer()
remove_punct_dict = {ord(punct):None for punct in string.punctuation}
def lem_token(tokens):
    return [lemmer.lemmatize(token) for token in tokens]
def lem_normalize(text):
    return lem_token(nltk.word_tokenize(text.lower().
↪ translate(remove_punct_dict)))
[7]: GREET_INPUT = ("hello,hi,sup")
GREET_RESPONSE = ['Hi',"Hello","I am glad you are talking to me!"]
def greet(sentence):
    for word in sentence.split():
        if word.lower() in GREET_INPUT:
            return random.choice(GREET_RESPONSE)
```

```
[8]: def response(user_response):
    sent_token.append(user_response)
    tfidfvec = TfidfVectorizer(tokenizer = lem_normalize,stop_words = 'english')
    tfidf = tfidfvec.fit_transform(sent_token)
    vals = cosine_similarity(tfidf[-1],tfidf)
    idx = vals.argsort()[0][-2]
    flat = vals.flatten()
    flat.sort()
    req_tfidf = flat[-2]
    sent_token.remove(user_response)
    if (req_tfidf == 0):
        return " I am Sorry! I dont understand you"
    else:
        return str(sent_token[idx])
```

```
[9]: flag = True
```

```

print("BOT : My Name is BOTHi, Let's Have Conversation! If you want to exit any_
↳time, just type Bye! ")
print("\n")
while (flag==True):
    user_response = input("You : ")

    if (user_response != "bye"):
        if (user_response == 'thanks'):
            flag = False
            print("BOT : You are welcome..")
        else:
            if (greet(user_response) != None):
                print("BOT : " + "\t" + greet(user_response))
            else:
                print("BOT L",end = "")
                print(response(user_response))
                print("\n")
            else:
                flag = False
                print("BOT : Goodbye! Take Care")
                BOT : My Name is BOTHi, Let's Have Conversation! If you want to exit any time,
                just type Bye!

```

You : hello
 BOT : I am glad you are talking to me!

You : what is python
 BOT L
 D:\Anaconda\Lib\site-packages\sklearn\feature_extraction\text.py:525:
 UserWarning: The parameter 'token_pattern' will not be used since 'tokenizer' is
 not None'
 warnings.warn(D:\Anaconda\Lib\site-
 packages\sklearn\feature_extraction\text.py:408: UserWarning: Your
 stop_words may be inconsistent with your preprocessing. Tokenizing the stop
 words generated tokens ['ha', 'le', 'u', 'wa'] not in
 stop_words.
 warnings.warn(
 I am Sorry! I dont understand you

You : what happend to you
 BOT L I am Sorry! I dont understand you

You : what
 BOT L I am Sorry! I dont understand you
 You : why
 BOT L I am Sorry! I dont understand you

You : have a lunch

BOT Land then i made lunch.

You : are you feel good

BOT Lgood!

You : then thank you

BOT Lthank you very much.

You : hi

BOT : Hello

You : bye

BOT : Goodbye! Take Care