CHATBOT DEPLOYMENT WITH FLASK AND JAVASCRIPT

Project url:

https://drive.google.com/drive/folders/1Op2qkOHuntHBXwJ6OWuKDjVrRaEDEjqy?usp=sharing

Working video:

https://drive.google.com/file/d/1AjiXHgfPMxDfg80r7cEI6Li0KP9CK3cU/view?usp=sharing

Initial Setup:

This repo currently contains the starter files.

Clone repo and create a virtual environment

```
$ cd chatbot-deployment
$ python3 -m venv venv
$ . venv/Scripts/activate
```

Install dependencies

```
$ (venv) pip install Flask torch torchvision nltk
```

Install nltk package

```
$ (venv) python
>>> import nltk
>>> nltk.download('punkt')
```

Modify `intents.json` with different intents and responses for your Chatbot

Run

```
$ (venv) python train.py
```

This will dump data.pth file. And then run the following command to test it in the console.

```
$ (venv) python chat.py
```

Kaggle link:

https://www.kaggle.com/datasets/grafstor/simple-dialogs-for-chatbot

- I have take the some data from the above kaggle text file and I have trained my chatbot and succuessfully implemented it
- After analysing the data set I have understood that the above kaggle file contains the data as sets in which each question has three different formats of presentation.
- This format is present throughout the data set
- A set contains three parts: { 11 questions, 11 questions, 10 questions }

Eg: A Set contains three parts(a,b,c)

Set a:

hi, how are you doing? i'm fine. how about yourself?

i'm fine. how about yourself? i'm pretty good. thanks for asking.

i'm pretty good. thanks for asking. no problem. so how have you been?

no problem. so how have you been? i've been great. what about you?

i've been great. what about you? i've been good. i'm in school right now.

i've been good. i'm in school right now. what school do you go to?

what school do you go to? i go to pcc.

i go to pcc. do you like it there?

do you like it there? it's okay. it's a really big campus.

it's okay. it's a really big campus. good luck with school.

good luck with school. thank you very much.

Set b:

how's it going? i'm doing well. how about you?

i'm doing well. how about you? never better, thanks.

never better, thanks. so how have you been lately?

so how have you been lately? i've actually been pretty good. you?

i've actually been pretty good. you? i'm actually in school right now.

i'm actually in school right now. which school do you attend?

which school do you attend? i'm attending pcc right now.

i'm attending pcc right now. are you enjoying it there?

are you enjoying it there? it's not bad. there are a lot of people there.

it's not bad. there are a lot of people there. good luck with that.

good luck with that. thanks.

Set c:

how are you doing today? i'm doing great. what about you?

i'm doing great. what about you? i'm absolutely lovely, thank you.

i'm absolutely lovely, thank you. everything's been good with you?

everything's been good with you? i haven't been better. how about yourself?

i haven't been better. how about yourself? i started school recently.

i started school recently. where are you going to school?

where are you going to school? i'm going to pcc.

i'm going to pcc.how do you like it so far?

how do you like it so far? i like it so far. my classes are pretty good right now.

i like it so far. my classes are pretty good right now. i wish you luck.

The given dataset is further converted into json format to further process it..

base.html

```
<!DOCTYPE html>
<html lang="en">
<link rel="stylesheet" href="{{ url_for('static', filename='style.css') }}">

<head>
    <meta charset="UTF-8">
    <title>Chatbot</title>
    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css
">
    <style>
        p{
            text-align: justify;
        }
    </style>
</head>
<body>
<div class="container">
    <div class="chatbox">
        <div class="chatbox__support">
            <div class="chatbox__header">
                <div class="chatbox__image--header">
                    <img src="https://img.icons8.com/color/48/000000/circled-
user-female-skin-type-5--v1.png" alt="image">
                </div>
                <div class="chatbox__content--header">
                    <h4 class="chatbox__heading--header">Chat</h4>
                    <p class="chatbox__description--header">Hi !, How can I
help you?</p>
                </div>
            </div>
            <div class="chatbox__messages">
                <div></div>
            </div>
            <div class="chatbox__footer">
                <input type="text" placeholder="Write a message...">
                <button class="chatbox__send--footer
send__button">Send</button>
            </div>
        </div>
        <div class="chatbox__button">
            <button><img src="{{ url_for('static', filename='images/chatbox-
icon.svg') }}" /></button>
        </div>
    </div>
</div>
```

```
<header class="jumbotron text-center">
    <h1>CHATBOT DEPLOYMENT WITH FLASK AND JAVASCRIPT</h1>
</header>

<div class="container">

    <section class="content">
        <h2>Natural Language Toolkit (NLTK)</h2>
        <p>
            Natural Language Toolkit (NLTK) is a comprehensive library in
Python designed to work with human language data, particularly in the fields
of natural language processing (NLP) and text analysis.
 </p>
    </section>

    <section class="content">
        <h2>TensorFlow</h2>
        <p>
            TensorFlow is an open-source machine learning framework developed
by Google that has gained widespread popularity for its versatility and
efficiency in building and training deep learning models.
        </p>
    </section>
</div>

<section class="additional-content container">
    <h2>Working Procedure</h2>
    <p>
        Serve only the Flask prediction API. The used HTML and JavaScript
files can be included in any Frontend application (with only a slight
modification) and can run completely separate from the Flask App then.
        Initial Setup:
    </p>
    <ul>
        <li>Clone repo and create a virtual environment</l1><br>
        <code>$ cd chatbot-deployment</code><br>

        <code>$ python3 -m venv venv</code><br>
        <code>$ .venv/Scripts/activate</code><br>
        <li>Install dependencies</li>
        <code>$ (venv) pip install Flask torch torchvision nltk</code>
        <li>Install NLTK package</li>
        <code>$ (venv) python</code><br>
        <code>>>> import nltk</code><br>
        <code>>>> nltk.download('punkt')</code><br>
        <li>Modify intents.json with different intents and responses for your
Chatbot</l1><br>
        <code>$ (venv) python train.py</code><br>
```

```
        <li>This will dump data.pth file. And then run the following command
to test it in the console.</li>
        <code>$ (venv) python chat.py</code>
    </ul>
</section>
<br>
<br>



    <script>
        $SCRIPT_ROOT = {{ request.script_root|tojson }};
    </script>
    <script type="text/javascript" src="{{ url_for('static',
filename='app.js') }}"></script>
    <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>
    <script
src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.5.3/dist/umd/popper.min.js"
></script>
    <script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"><
/script>

</body>
</html>
```

train.py

```
import numpy as np
import random
import json

import torch
import torch.nn as nn
from torch.utils.data import Dataset, DataLoader

from nltk_utils import bag_of_words, tokenize, stem
from model import NeuralNet

with open('intents.json', 'r') as f:
    intents = json.load(f)

all_words = []
tags = []
xy = []
for intent in intents['intents']:
    tag = intent['tag']
    tags.append(tag)
```

```
    for pattern in intent['patterns']:
        w = tokenize(pattern)
        all_words.extend(w)
        xy.append((w, tag))
ignore_words = ['?', '.', '!']
all_words = [stem(w) for w in all_words if w not in ignore_words]
all_words = sorted(set(all_words))
tags = sorted(set(tags))

print(len(xy), "patterns")
print(len(tags), "tags:", tags)
print(len(all_words), "unique stemmed words:", all_words)

X_train = []
y_train = []
for (pattern_sentence, tag) in xy:
    bag = bag_of_words(pattern_sentence, all_words)
    X_train.append(bag)
    label = tags.index(tag)
    y_train.append(label)

X_train = np.array(X_train)
y_train = np.array(y_train)

num_epochs = 1000
batch_size = 8
learning_rate = 0.001
input_size = len(X_train[0])
hidden_size = 8
output_size = len(tags)
print(input_size, output_size)

class ChatDataset(Dataset):

    def __init__(self):
        self.n_samples = len(X_train)
        self.x_data = X_train
        self.y_data = y_train

    def __getitem__(self, index):
        return self.x_data[index], self.y_data[index]

    def __len__(self):
        return self.n_samples

dataset = ChatDataset()
train_loader = DataLoader(dataset=dataset,
                          batch_size=batch_size,
```

```
                          shuffle=True,
                          num_workers=0)

device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')

model = NeuralNet(input_size, hidden_size, output_size).to(device)

criterion = nn.CrossEntropyLoss()
optimizer = torch.optim.Adam(model.parameters(), lr=learning_rate)

for epoch in range(num_epochs):
    for (words, labels) in train_loader:
        words = words.to(device)
        labels = labels.to(dtype=torch.long).to(device)

        outputs = model(words)
        loss = criterion(outputs, labels)

        optimizer.zero_grad()
        loss.backward()
        optimizer.step()

    if (epoch+1) % 100 == 0:
        print (f'Epoch [{epoch+1}/{num_epochs}], Loss: {loss.item():.4f}')


print(f'final loss: {loss.item():.4f}')

data = {
"model_state": model.state_dict(),
"input_size": input_size,
"hidden_size": hidden_size,
"output_size": output_size,
"all_words": all_words,
"tags": tags
}

FILE = "data.pth"
torch.save(data, FILE)

print(f'training complete. file saved to {FILE}')
```

nltk_utils.py

```
import numpy as np
import nltk
# nltk.download('punkt')
```

```
from nltk.stem.porter import PorterStemmer
stemmer = PorterStemmer()


def tokenize(sentence):
    """
    split sentence into array of words/tokens
    a token can be a word or punctuation character, or number
    """
    return nltk.word_tokenize(sentence)


def stem(word):
    """
    stemming = find the root form of the word
    examples:
    words = ["organize", "organizes", "organizing"]
    words = [stem(w) for w in words]
    -> ["organ", "organ", "organ"]
    """
    return stemmer.stem(word.lower())


def bag_of_words(tokenized_sentence, words):
    """
    return bag of words array:
    1 for each known word that exists in the sentence, 0 otherwise
    example:
    sentence = ["hello", "how", "are", "you"]
    words = ["hi", "hello", "I", "you", "bye", "thank", "cool"]
    bog   = [  0 ,    1 ,    0 ,   1 ,    0 ,    0 ,      0]
    """

    sentence_words = [stem(word) for word in tokenized_sentence]
    bag = np.zeros(len(words), dtype=np.float32)
    for idx, w in enumerate(words):
        if w in sentence_words:
            bag[idx] = 1

    return bag
```

model.py

```
import torch
import torch.nn as nn
class NeuralNet(nn.Module):
    def __init__(self, input_size, hidden_size, num_classes):
        super(NeuralNet, self).__init__()
```

```
        self.l1 = nn.Linear(input_size, hidden_size)
        self.l2 = nn.Linear(hidden_size, hidden_size)
        self.l3 = nn.Linear(hidden_size, num_classes)
        self.relu = nn.ReLU()

    def forward(self, x):
        out = self.l1(x)
        out = self.relu(out)
        out = self.l2(out)
        out = self.relu(out)
        out = self.l3(out)
        return out
```

chat.py

```
import random
import json

import torch

from model import NeuralNet
from nltk_utils import bag_of_words, tokenize

device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')

with open('intents.json', 'r') as json_data:
    intents = json.load(json_data)

FILE = "data.pth"
data = torch.load(FILE)

input_size = data["input_size"]
hidden_size = data["hidden_size"]
output_size = data["output_size"]
all_words = data['all_words']
tags = data['tags']
model_state = data["model_state"]

model = NeuralNet(input_size, hidden_size, output_size).to(device)
model.load_state_dict(model_state)
model.eval()

bot_name = "Vishnu"

def get_response(msg):
    sentence = tokenize(msg)
    X = bag_of_words(sentence, all_words)
```

```
    X = X.reshape(1, X.shape[0])
    X = torch.from_numpy(X).to(device)

    output = model(X)
    _, predicted = torch.max(output, dim=1)

    tag = tags[predicted.item()]

    probs = torch.softmax(output, dim=1)
    prob = probs[0][predicted.item()]
    if prob.item() > 0.75:
        for intent in intents['intents']:
            if tag == intent["tag"]:
                return random.choice(intent['responses'])

    return "I do not understand..."


if __name__ == "__main__":
    print("Let's chat! (type 'quit' to exit)")
    while True:
        sentence = input("You: ")
        if sentence == "quit":
            break

        resp = get_response(sentence)
        print(resp)
```

app.py

```
from flask import Flask, render_template, request, jsonify

from chat import get_response

app = Flask(__name__)

@app.get("/")
def index_get():
    return render_template("base.html")

@app.post("/predict")
def predict():
    text = request.get_json().get("message")
    # TODO: check if text is valid
    response = get_response(text)
    message = {"answer": response}
    return jsonify(message)
```

```
if __name__ == "__main__":
    app.run(debug=True)
```

intents.json

```
{
  "intents": [
    {
      "tag": "doing",
      "patterns": [
       "hi, how are you doing?","how's it going?","how are you doing today?"
      ],
      "responses": [
      "i'm fine. how about yourself?","i'm doing well. how about you?","i'm
doing great. what about you?"
      ]
    },
    {
      "tag": "good",
      "patterns": [
       "i'm pretty good. thanks for asking.","never better, thanks.","i'm
absolutely lovely, thank you."
      ],
      "responses": [
      "no problem. so how have you been?","so how have you been
lately?","everything's been good with you?"
      ]
    },
    {
      "tag": "about",
      "patterns": [
       "i've been great. what about you?","i've actually been pretty good.
you?","i haven't been better. how about yourself?"
      ],
      "responses": [
      "i've been good. i'm in school right now.","i'm actually in school right
now.","i started school recently."
      ]
    },
    {
      "tag": "school",
      "patterns": [
       "what school do you go to?","which school do you attend?","where are
you going to school?"
      ],
      "responses": [
      "i go to pcc.","i'm attending pcc right now.","i'm going to pcc."
```

```
      ]
    },
    {
      "tag": "like",
      "patterns": [
       "do you like it there?","are you enjoying it there?","how do you like
it so far?"
      ],
      "responses": [
      "it's okay. it's a really big campus.","it's not bad. there are a lot of
people there.","i like it so far. my classes are pretty good right now."
      ]
    },
    {
      "tag": "luck",
      "patterns": [
       "good luck with school.","good luck with that.","i wish you luck."
      ],
      "responses": [
      "thank you very much.","thanks."
      ]
    },
    {
      "tag": "today",
      "patterns": [
       "it's an ugly day today.","it doesn't look very nice outside today.","i
wish it was a nicer day today."
      ],
      "responses": [
      "i know. i think it may rain.","you're right. i think it's going to rain
later.","that is true. i hope it doesn't rain."
      ]
    },
    {
      "tag": "summer",
      "patterns": [
       "it's the middle of summer, it shouldn't rain today.","in the middle of
the summer, it shouldn't be raining.","it wouldn't rain in the middle of the
summer."
      ],
      "responses": [
      "that would be weird.","that wouldn't seem right.","it wouldn't seem
right if it started raining right now."
      ]
    },
    {
      "tag": "ninety",
      "patterns": [
```

```
       "yeah, especially since it's ninety degrees outside.","considering that
it's over ninety degrees outside, that would be weird.","it would be weird if
it started raining in ninety degree weather."
      ],
      "responses": [
      "i know, it would be horrible if it rained and it was hot
outside.","exactly, it wouldn't be nice if it started raining. it's too
hot.","any rain right now would be pointless."
      ]
    },
    {
      "tag": "right",
      "patterns": [
       "yes, it would be.","i know, you're absolutely right.","that's right,
it really would be."
      ],
      "responses": [
      "i really wish it wasn't so hot every day. ","i wish it would cool off
one day.","i want it to cool down some."
      ]
    },
    {
      "tag": "winter",
      "patterns": [
       "me too. i can't wait until winter.","that's how i feel, i want winter
to come soon.","i know what you mean, i can't wait until it's winter."
      ],
      "responses": [
      "i like winter too, but sometimes it gets too cold.","i enjoy the
winter, but it gets really cold sometimes.","winter is great. i wish it didn't
get so cold sometimes though."
      ]
    },
    {
      "tag": "cold/hot",
      "patterns": [
       "i'd rather be cold than hot.","i know what you mean, but i'd rather be
cold than hot.","i would rather deal with the winter than the summer."
      ],
      "responses": [
      "me too.","that's exactly how i feel."
      ]
    },
    {
      "tag": "nice",
      "patterns": [
       "it's such a nice day.","isn't it a nice day?","don't you think it's
nice out?"
```

```
      ],
      "responses": [
      "yes, it is.","it really is.","yes, i think so too."
      ]
    },
    {
      "tag": "rain",
      "patterns": [
       "it looks like it may rain soon.","it seems that it may rain today.","i
think that it's going to rain."
      ],
      "responses": [
      "yes, and i hope that it does.","hopefully it will.","i hope that it
does rain."
      ]
    },
    {
      "tag": "come",
      "patterns": [
       "why is that?","how come?","you like the rain?"
      ],
      "responses": [
      "i really love how rain clears the air.","i like how clear the sky gets
after it rains.","the sky looks so clean after it rains. i love it."
      ]
    },

     {
      "tag": "smells",
      "patterns": [
       "me too. it always smells so fresh after it rains.","i feel the same
way. it smells so good after it rains.","i understand. rain does make it smell
cleaner."
      ],
      "responses": [
      "yes, but i love the night air after it rains.","i especially love the
night air when it rains.","i love most how it is at night after it rains."
      ]
    }, {
      "tag": "really",
      "patterns": [
       "really? why is it?","really? why?","how come?"
      ],
      "responses": [
      "because you can see the stars perfectly.","the stars look so much
closer after it rains.","you can see the stars so much more clearly after it
rains."
      ]
```

```
    },
    {
      "tag": "rains",
      "patterns": [
       "i really hope it rains today.","i really want it to rain today.","i
would love for it to rain today."
      ],
      "responses": [
      "yeah, me too.","yeah, so do i."
      ]
    },
    {
      "tag": "weekend",
      "patterns": [
       "i really want to go to the beach this weekend.","i would like to take
a trip to the beach this weekend.","it would be nice to go to the beach
sometime this weekend."
      ],
      "responses": [
      "that sounds like fun. what's the weather going to be like?","a trip to
the beach would be fun. how is the weather going to be?","what's the weather
going to be like? i may want to go too."
      ]
    },
    {
      "tag": "weather",
      "patterns": [
       "i heard that it's going to be warm this weekend.","the forecast says
that it will be warm on the weekend.","the weather this weekend is supposed to
be warm."
      ],
      "responses": [
      "is it going to be perfect beach weather?","so do you think it'll be
perfect weather for the beach?","will it be good beach weather?"
      ]
    },
    {
      "tag": "sounds",
      "patterns": [
       "i believe so.","it sounds like it will be.","i think it will be."
      ],
      "responses": [
      "good. i hope it doesn't cool off this weekend.","i really hope it
doesn't get cold.","it wouldn't be good if it got cold this weekend."
      ]
    },
    {
      "tag": "beach",
```

```
      "patterns": [
       "i know. i really want to go to the beach.","that would ruin things, i
want to go so badly.","i want this trip to be perfect, i hope it stays warm."
      ],
      "responses": [
      "but you know that california weather is really unpredictable.","the
weather in california is unpredictable, so you never know.","this california
weather is so uncertain, it's impossible to know what'll happen."
      ]
    },
    {
      "tag": "minute",
      "patterns": [
       "you're right. one minute it's hot, and then the next minute it's
cold.","that is true. the weather is constantly changing.","i know. every day
the weather seems different."
      ],
      "responses": [
      "i really wish the weather would just stay the same.","it would be nice
if the weather would never change.","i would love it if it wasn't always so
unpredictable."
      ]
    },
    {
      "tag": "activities",
      "patterns": [
       "i do too. that way we can have our activities planned ahead of
time.","that would be great, then we could plan things sooner.","that would
make it easier for us to make plans."
      ],
      "responses": [
      "yeah, that would make things a lot easier.","true. predictable weather
would make life easier."
      ]
    },
    {
      "tag": "alice",
      "patterns": [
       "hello, may i speak to alice please?","hi, how are you. is alice
there?","is alice available?"
      ],
      "responses": [
      "this is she. how's it going?","speaking. what's up?","you're talking to
her."
      ]
    },
    {
      "tag": "call",
```

```
      "patterns": [
       "i've been trying to call you all day.","why haven't you answered the
phone?","i've called you a hundred times today."
      ],
      "responses": [
      "sorry about that. i was cleaning up.","my bad, i had chores to do.","i
was busy doing something. i apologize."
      ]
    },
    {
      "tag": "okay",
      "patterns": [
       "it's okay.","that's all right.","no problem."
      ],
      "responses": [
      "so what were you calling me about?","what was the reason for your
call?","did you need something?"
      ]
    },
    {
      "tag": "tomorrow",
      "patterns": [
       "oh, i just wanted to see if you wanted to hang out tomorrow.","i want
to do something tomorrow with you.","do you want to do something tomorrow?"
      ],
      "responses": [
      "sure, what did you want to do?","sounds good. what did you have in
mind?","is there somewhere special you wanted to go?"
      ]
    },
    {
      "tag": "movie",
      "patterns": [
       "maybe we can go see a movie or something.","i was thinking about
seeing a movie.","how about a movie?"
      ],
      "responses": [
      "that sounds like fun. let's do it.","okay, let's go see a movie. ","a
movie sounds good."
      ]
    },
    {
      "tag": "then",
      "patterns": [
       "i'll see you tomorrow then.","until then.","call me tomorrow then."
      ],
      "responses": [
      "see you then. goodbye.","talk to you later."
```

```
      ]
    },
    {
      "tag": "girl",
      "patterns": ["have you seen the new girl in school?","there's a new girl
in school, have you seen her yet?","have you met the new girl?"
      ],
      "responses": ["no, i haven't","i haven't seen her yet.","no. have you?"

      ]
    },
    {
      "tag": "pretty",
      "patterns": ["she's really pretty.","i think that she is very
pretty.","she's one of the prettiest girls at the school."

      ],
      "responses": ["describe her to me.","tell me how she looks.","what does
she look like?"

      ]
    },
    {
      "tag": "short",
      "patterns": ["she's not too tall.","she's kind of short.","well, she's
quite short."

      ],
      "responses": ["well, how tall is she?","what height is she?","how tall
would you say that she is?"

      ]
    },
    {
      "tag": "five",
      "patterns": ["she's about five feet even.","she's probably about five
feet.","i would say she's only five feet."

      ],
      "responses": ["what does she look like, though?","that's nice, but tell
me what she looks like","what about her facial features?"

      ]
    },
    {
      "tag": "brown",
```

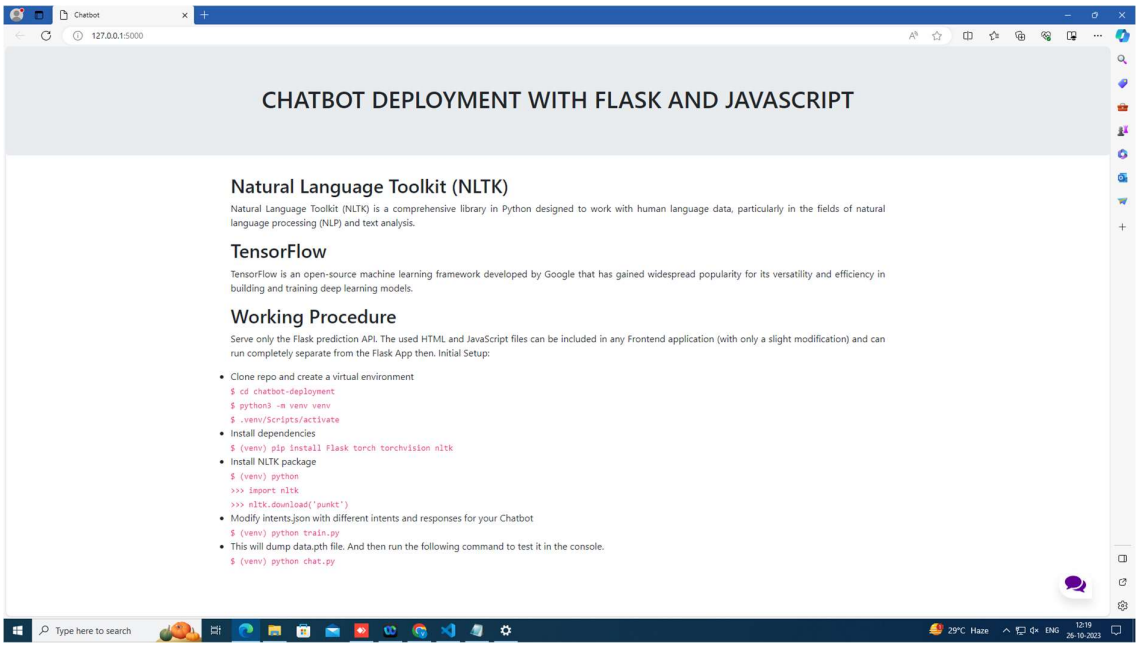
```
      "patterns": ["she has pretty light brown eyes.","the first thing i
noticed was her beautiful brown eyes.","she has light brown eyes, absolutely
beautiful."

      ],
      "responses": ["i may know which girl you're talking about.","i think i
might've bumped into her before.","i think i know who you're talking about."

      ]
    },
    {
      "tag": "seen",
      "patterns": ["so you have seen her around?","are you telling me that
you've seen her before?","have you seen her?"

      ],
      "responses": ["yes, i have.","i believe so."

      ]
    },

    {
      "tag": "yesterday",
      "patterns": ["why weren't you at school yesterday?","what reason do you
have for missing school?","why didn't you go to school yesterday?"

      ],
      "responses": ["i wasn't really feeling well.","i was sick.","i stayed
home because i wasn't feeling well."

      ]
    },
    {
      "tag": "what",
      "patterns": ["what was wrong with you?","how were you sick?","what was
your problem?"

      ],
      "responses": ["my stomach was upset.","i had a stomachache.","my stomach
was bothering me."

      ]
    },
    {
      "tag": "better",
      "patterns": ["do you feel better now?","did it get any better?","are you
feeling any better?"
```

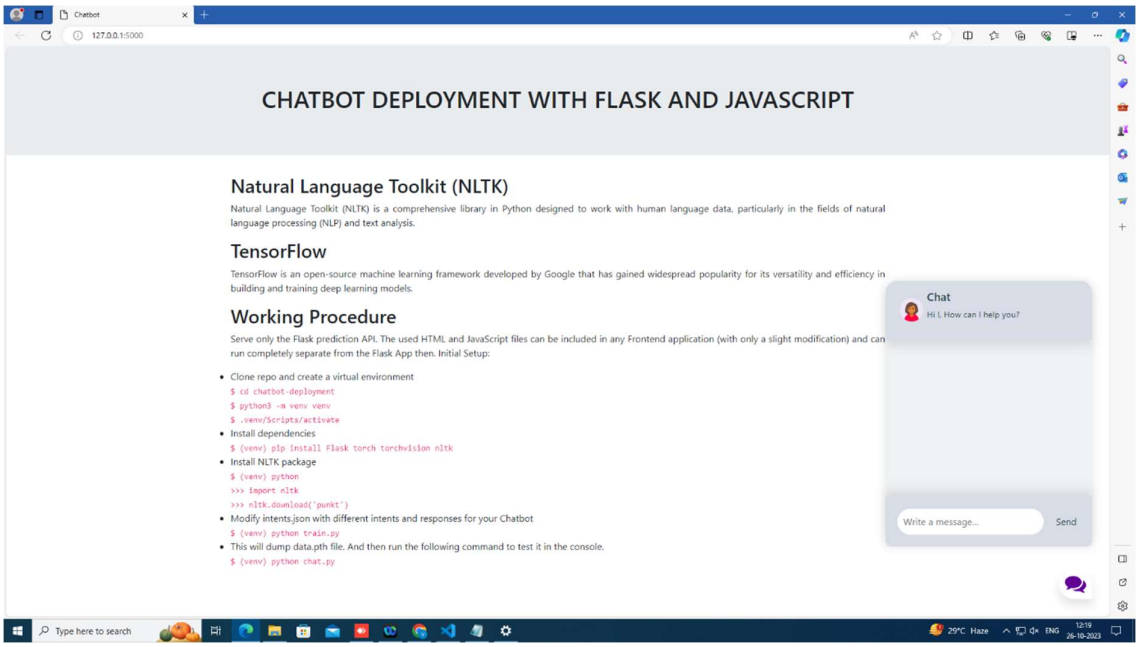
```
      ],
      "responses": ["i don't really feel too well yet.","i'm still feeling
under the weather.","i'm still feeling a little sick."

      ]
    },

    {
      "tag": "anything",
      "patterns": ["do you want anything to make you feel better?","would you
like anything for your stomach?","i'm going to the store, would you like any
pepto bismol?"

      ],
      "responses": ["no, thanks. i already took some medicine.","i took
something earlier.","that's okay."

      ]
    },
    {
      "tag": "feel",
      "patterns": ["i hope you feel better","get better."

      ],
      "responses": ["thank you.","thanks a lot."

      ]
    },
    {
      "tag": "favorite",
      "patterns": [
       "what's your favorite movie?","which movie is your favorite to
watch?","out of every movie that you've seen, which one is your favorite?"
      ],
      "responses": [
      "my favorite movie is superbad.","i have to say, my favorite movie is
superbad.","i'm going to have to say that superbad is the best movie ever."
      ]
    },
    {
      "tag": "why",
      "patterns": [
       "oh, why is that?","is that right? why?","you think so, how come?"
      ],
      "responses": [
      "it's the funniest movie that i've ever seen.","honestly, it is one of
the funniest movies i've seen in a long time.","well, superbad is super
funny."
```

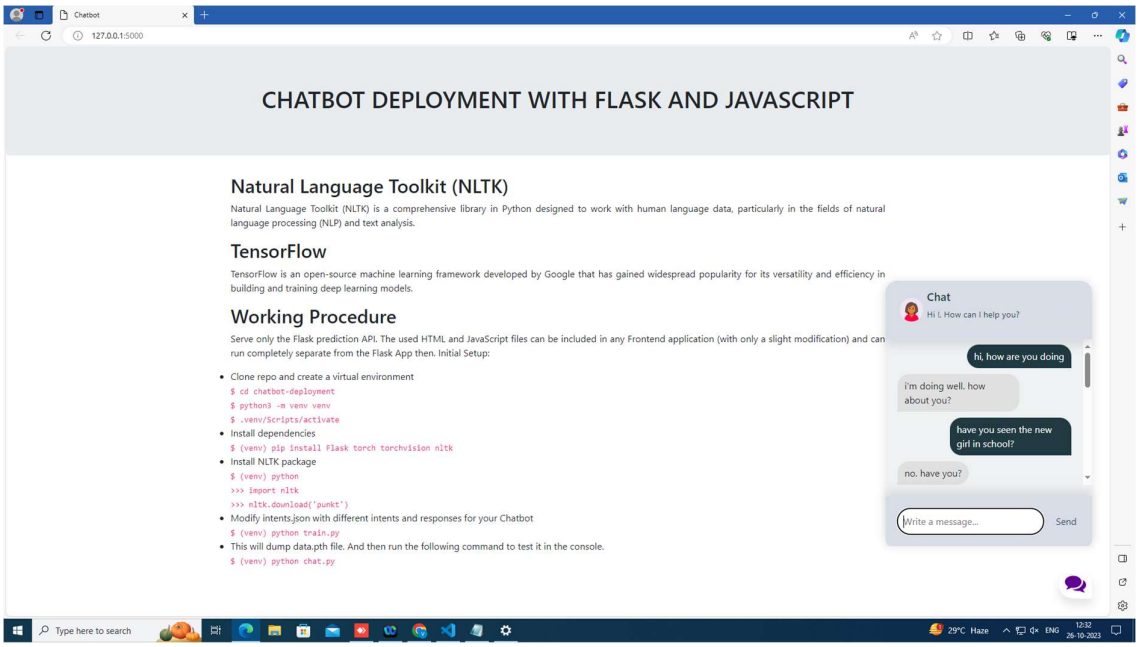
```
      ]
    },
    {
      "tag": "funny",
      "patterns": [
       "that's true. it is a very funny movie.","you're right. that movie is
hilarious.","you're not lying, i found that movie absolutely hilarious."
      ],
      "responses": [
      "you've seen it before?","i didn't think you saw that movie.","i didn't
know that you saw superbad before."
      ]
    }
    ,
    {
      "tag": "theaters",
      "patterns": [
       "yes, i saw that movie the first day it came out in theaters.","i went
to see it the day it came out.","i made sure to be in line to see it the first
day it came out."
      ],
      "responses": [
      "didn't you laugh through the whole movie?","i was laughing through the
whole movie.","i couldn't keep from laughing throughout the whole movie."
      ]
    }
    ,
    {
      "tag": "tears",
      "patterns": [
       "me too. that movie brought tears to my eyes.","i couldn't help
laughing, either.","i was laughing hysterically the whole time; my stomach
muscles hurt afterwards."
      ],
      "responses": [
      "mine too.","same here.","that's exactly how i felt."
      ]
    }
    ,
    {
      "tag": "bought",
      "patterns": [
       "i have it on dvd at my house if you want to come over and watch
it.","i bought the movie. would you like to come to my house and watch it?","i
got the movie when it came out on dvd, do you want to come over?"
      ],
      "responses": [
      "sure, let's go.","of course."
```

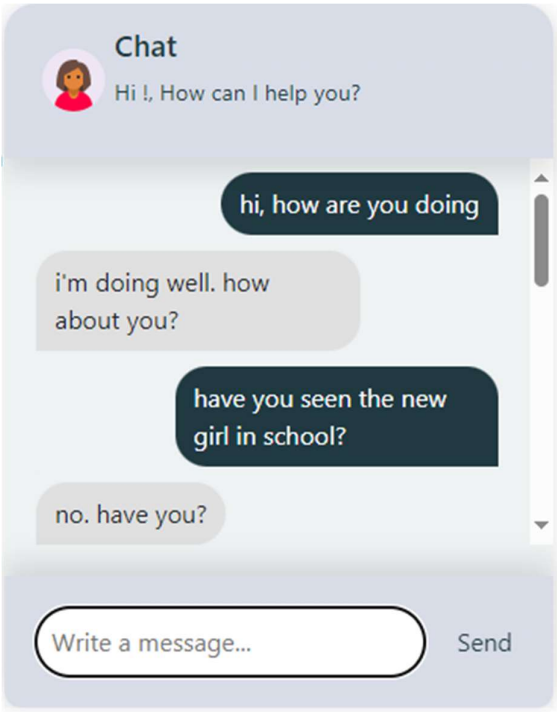
```
        ]
    }


  ]
}
```

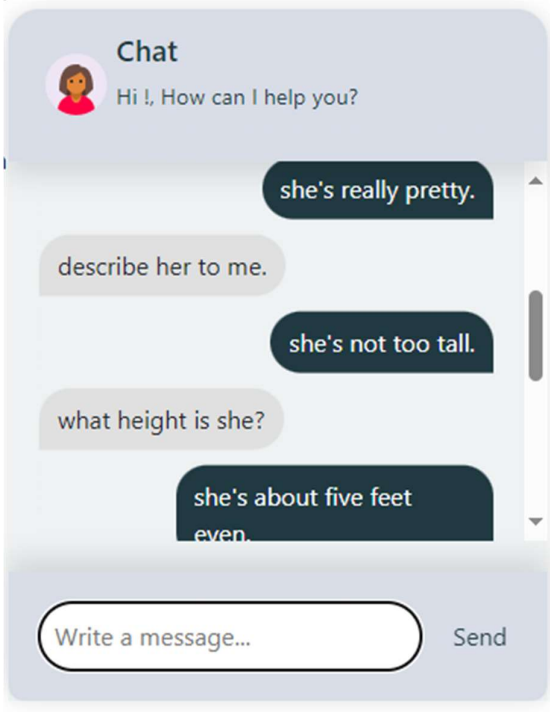
Output:

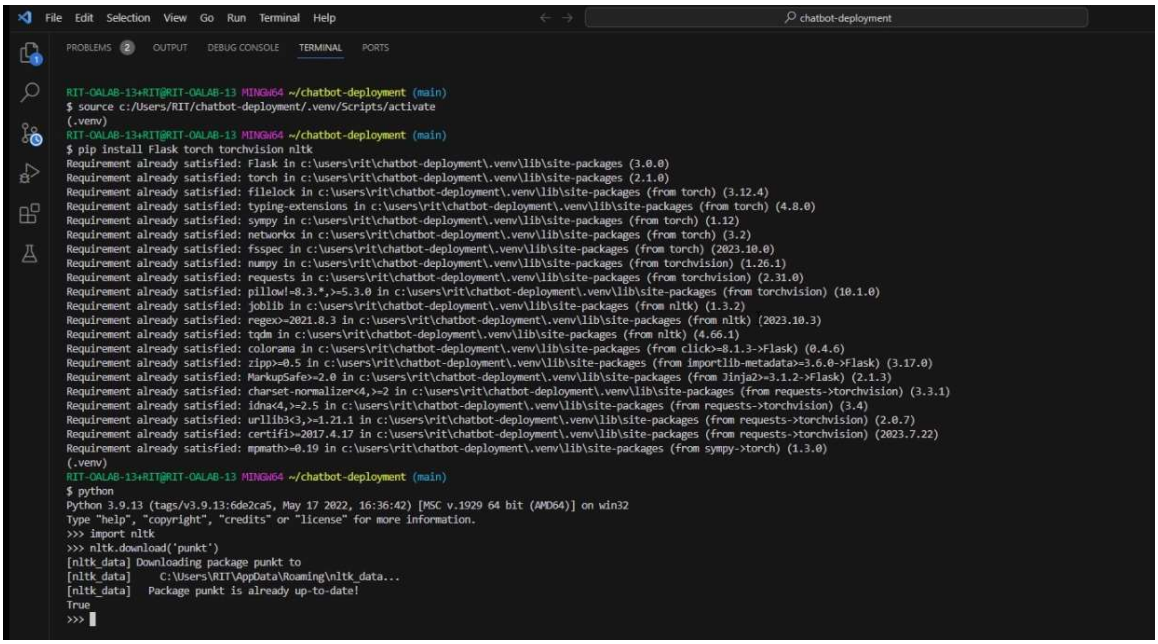

Chatbot
127.0.0.1:5000
CHATBOT DEPLOYMENT WITH FLASK AND JAVASCRIPT
Natural Language Toolkit (NLTK)
Natural Language Toolkit (NLTK) is a comprehensive library in Python designed to work with human language data, particularly in the fields of natural language processing (NLP) and text analysis.
TensorFlow
TensorFlow is an open-source machine learning framework developed by Google that has gained widespread popularity for its versatility and efficiency in building and training deep learning models.
Working Procedure
Serve only the Flask prediction API. The used HTML and JavaScript files can be included in any Frontend application (with only a slight modification) and can run completely separate from the Flask App then. Initial Setup:
Clone repo and create a virtual environment
$ cd chatbot-deployment
$ python3 -m venv venv
$ .venv/Scripts/activate
Install dependencies
$ (venv) pip install Flask torch torchvision nltk
Install NLTK package
$ (venv) python
>>> import nltk
>>> nltk.download('punkt')
Modify intents.json with different intents and responses for your Chatbot
$ (venv) python train.py
This will dump data.pth file. And then run the following command to test it in the console.
$ (venv) python chat.py
Type here to search
29°C Haze
ENG
12:19
26-10-2023

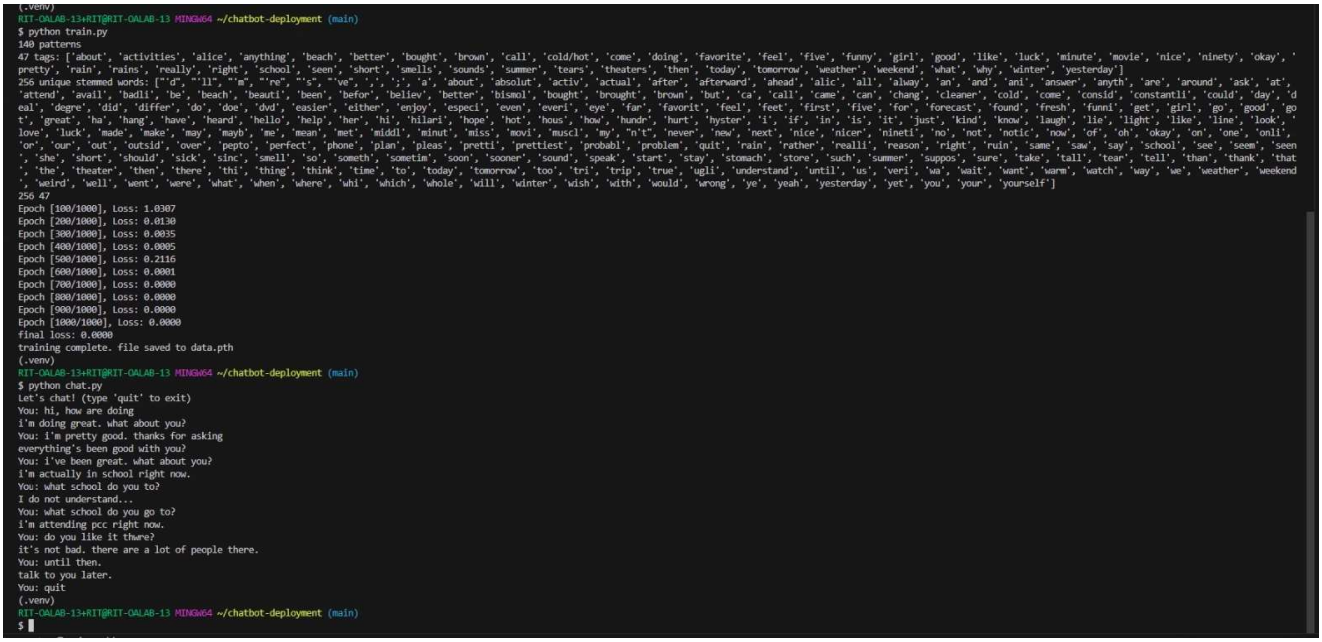

Chatbot
127.0.0.1:5000
CHATBOT DEPLOYMENT WITH FLASK AND JAVASCRIPT
Natural Language Toolkit (NLTK)
Natural Language Toolkit (NLTK) is a comprehensive library in Python designed to work with human language data, particularly in the fields of natural language processing (NLP) and text analysis.
TensorFlow
TensorFlow is an open-source machine learning framework developed by Google that has gained widespread popularity for its versatility and efficiency in building and training deep learning models.
Working Procedure
Serve only the Flask prediction API. The used HTML and JavaScript files can be included in any Frontend application (with only a slight modification) and can run completely separate from the Flask App then. Initial Setup:
Clone repo and create a virtual environment
$ cd chatbot-deployment
$ python3 -m venv venv
$ .venv/Scripts/activate
Install dependencies
$ (venv) pip install Flask torch torchvision nltk
Install NLTK package
$ (venv) python
>>> import nltk
>>> nltk.download('punkt')
Modify intents.json with different intents and responses for your Chatbot
$ (venv) python train.py
This will dump data.pth file. And then run the following command to test it in the console.
$ (venv) python chat.py
Chat
Hi I, How can I help you?
Write a message...
Send
Type here to search
29°C Haze
ENG
12:19
26-10-2023


Chatbot
127.0.0.1:5000
CHATBOT DEPLOYMENT WITH FLASK AND JAVASCRIPT
Natural Language Toolkit (NLTK)
Natural Language Toolkit (NLTK) is a comprehensive library in Python designed to work with human language data, particularly in the fields of natural language processing (NLP) and text analysis.
TensorFlow
TensorFlow is an open-source machine learning framework developed by Google that has gained widespread popularity for its versatility and efficiency in building and training deep learning models.
Working Procedure
Serve only the Flask prediction API. The used HTML and JavaScript files can be included in any Frontend application (with only a slight modification) and can run completely separate from the Flask App then. Initial Setup:
Clone repo and create a virtual environment
$ cd chatbot-deployment
$ python3 -m venv venv
$ .venv/Scripts/activate
Install dependencies
$ (venv) pip install Flask torch torchvision nltk
Install NLTK package
$ (venv) python
>>> import nltk
>>> nltk.download('punkt')
Modify intents.json with different intents and responses for your Chatbot
$ (venv) python train.py
This will dump data.pth file. And then run the following command to test it in the console.
$ (venv) python chat.py
Chat
Hi !, How can I help you?
hi, how are you doing
i'm doing well. how about you?
have you seen the new girl in school?
no. have you?
Write a message...
Send
Type here to search
29°C Haze
ENG
12:32
26-10-2023


Chat
Hi !, How can I help you?
hi, how are you doing
i'm doing well. how about you?
have you seen the new girl in school?
no. have you?
Write a message...
Send


Chat
Hi !, How can I help you?
she's really pretty.
describe her to me.
she's not too tall.
what height is she?
she's about five feet
Write a message...
Send


File Edit Selection View Go Run Terminal Help
chatbot-deployment
PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS
RIT-OALAB-13+RIT@RIT-OALAB-13 MINGW64 ~/chatbot-deployment (main)
$ source c:/Users/RIT/chatbot-deployment/.venv/Scripts/activate
(.venv)
RIT-OALAB-13+RIT@RIT-OALAB-13 MINGW64 ~/chatbot-deployment (main)
$ pip install Flask torch torchvision nltk
Requirement already satisfied: Flask in c:\users\rit\chatbot-deployment\.venv\lib\site-packages (3.0.0)
Requirement already satisfied: torch in c:\users\rit\chatbot-deployment\.venv\lib\site-packages (2.1.0)
Requirement already satisfied: filelock in c:\users\rit\chatbot-deployment\.venv\lib\site-packages (from torch) (3.12.4)
Requirement already satisfied: typing-extensions in c:\users\rit\chatbot-deployment\.venv\lib\site-packages (from torch) (4.8.0)
Requirement already satisfied: sympy in c:\users\rit\chatbot-deployment\.venv\lib\site-packages (from torch) (1.12)
Requirement already satisfied: networkx in c:\users\rit\chatbot-deployment\.venv\lib\site-packages (from torch) (3.2)
Requirement already satisfied: fsspec in c:\users\rit\chatbot-deployment\.venv\lib\site-packages (from torch) (2023.10.0)
Requirement already satisfied: numpy in c:\users\rit\chatbot-deployment\.venv\lib\site-packages (from torchvision) (1.26.1)
Requirement already satisfied: requests in c:\users\rit\chatbot-deployment\.venv\lib\site-packages (from torchvision) (2.31.0)
Requirement already satisfied: pillow!=8.3.*,>=5.3.0 in c:\users\rit\chatbot-deployment\.venv\lib\site-packages (from torchvision) (10.1.0)
Requirement already satisfied: joblib in c:\users\rit\chatbot-deployment\.venv\lib\site-packages (from nltk) (1.3.2)
Requirement already satisfied: regex>=2021.8.3 in c:\users\rit\chatbot-deployment\.venv\lib\site-packages (from nltk) (2023.10.3)
Requirement already satisfied: tqdm in c:\users\rit\chatbot-deployment\.venv\lib\site-packages (from nltk) (4.66.1)
Requirement already satisfied: colorama in c:\users\rit\chatbot-deployment\.venv\lib\site-packages (from click>=8.1.3->Flask) (0.4.6)
Requirement already satisfied: zipp>=0.5 in c:\users\rit\chatbot-deployment\.venv\lib\site-packages (from importlib-metadata>=3.6.0->Flask) (3.17.0)
Requirement already satisfied: MarkupSafe>=2.0 in c:\users\rit\chatbot-deployment\.venv\lib\site-packages (from Jinja2>=3.1.2->Flask) (2.1.3)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\users\rit\chatbot-deployment\.venv\lib\site-packages (from requests->torchvision) (3.3.1)
Requirement already satisfied: idna<4,>=2.5 in c:\users\rit\chatbot-deployment\.venv\lib\site-packages (from requests->torchvision) (3.4)
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\users\rit\chatbot-deployment\.venv\lib\site-packages (from requests->torchvision) (2.0.7)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\rit\chatbot-deployment\.venv\lib\site-packages (from requests->torchvision) (2023.7.22)
Requirement already satisfied: mpmath>=0.19 in c:\users\rit\chatbot-deployment\.venv\lib\site-packages (from sympy->torch) (1.3.0)
(.venv)
RIT-OALAB-13+RIT@RIT-OALAB-13 MINGW64 ~/chatbot-deployment (main)
$ python
Python 3.9.13 (tags/v3.9.13:6de2ca5, May 17 2022, 16:36:42) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import nltk
>>> nltk.download('punkt')
[nltk_data] Downloading package punkt to
[nltk_data] C:\Users\RIT\AppData\Roaming\nltk_data...
[nltk_data] Package punkt is already up-to-date!
True
>>>


(.venv)
RIT-OALAB-13+RIT@RIT-OALAB-13 MINGW64 ~/chatbot-deployment (main)
$ python train.py
140 patterns
47 tags: ['about', 'activities', 'alice', 'anything', 'beach', 'better', 'bought', 'brown', 'call', 'cold/hot', 'come', 'doing', 'favorite', 'feel', 'five', 'funny', 'girl', 'good', 'like', 'luck', 'minute', 'movie', 'nice', 'ninety', 'okay', 'pretty', 'rain', 'rains', 'really', 'right', 'school', 'seen', 'short', 'smells', 'sounds', 'summer', 'tears', 'theaters', 'then', 'today', 'tomorrow', 'weather', 'weekend', 'what', 'why', 'winter', 'yesterday']
256 unique stemmed words: ["'d", "'ll", "'m", "'re", "'s", "'ve", ',', '.', 'a', 'about', 'absolut', 'activ', 'actual', 'after', 'afterward', 'ahead', 'alic', 'all', 'alway', 'an', 'and', 'ani', 'answer', 'anyth', 'are', 'around', 'ask', 'at', 'attend', 'avail', 'badli', 'be', 'beach', 'beauti', 'been', 'befor', 'believ', 'better', 'bismol', 'bought', 'brought', 'brown', 'but', 'ca', 'call', 'came', 'can', 'chang', 'cleaner', 'cold', 'come', 'consid', 'constantli', 'could', 'day', 'deal', 'degre', 'did', 'differ', 'do', 'doe', 'dvd', 'easier', 'either', 'enjoy', 'especi', 'even', 'everi', 'eye', 'far', 'favorit', 'feel', 'feet', 'first', 'five', 'for', 'forecast', 'found', 'fresh', 'funni', 'get', 'girl', 'go', 'good', 'got', 'great', 'ha', 'hang', 'have', 'heard', 'hello', 'help', 'her', 'hi', 'hilari', 'hope', 'hot', 'hous', 'how', 'hundr', 'hurt', 'hyster', 'i', 'if', 'in', 'is', 'it', 'just', 'kind', 'know', 'laugh', 'lie', 'light', 'like', 'line', 'look', 'love', 'luck', 'made', 'make', 'may', 'mayb', 'me', 'mean', 'met', 'middl', 'minut', 'miss', 'movi', 'muscl', 'my', "n't", 'never', 'new', 'next', 'nice', 'nicer', 'nineti', 'no', 'not', 'notic', 'now', 'of', 'oh', 'okay', 'on', 'one', 'onli', 'or', 'our', 'out', 'outsid', 'over', 'pepto', 'perfect', 'phone', 'plan', 'pleas', 'pretti', 'prettiest', 'probabl', 'problem', 'quit', 'rain', 'rather', 'realli', 'reason', 'right', 'ruin', 'same', 'saw', 'say', 'school', 'see', 'seem', 'seen', 'she', 'short', 'should', 'sick', 'sinc', 'smell', 'so', 'someth', 'sometim', 'soon', 'sooner', 'sound', 'speak', 'start', 'stay', 'stomach', 'store', 'such', 'summer', 'suppos', 'sure', 'take', 'tall', 'tear', 'tell', 'than', 'thank', 'that', 'the', 'theater', 'then', 'there', 'thi', 'thing', 'think', 'time', 'to', 'today', 'tomorrow', 'too', 'tri', 'trip', 'true', 'ugli', 'understand', 'until', 'us', 'veri', 'wa', 'wait', 'want', 'warm', 'watch', 'way', 'we', 'weather', 'weekend', 'weird', 'well', 'went', 'were', 'what', 'when', 'where', 'whi', 'which', 'whole', 'will', 'winter', 'wish', 'with', 'would', 'wrong', 'ye', 'yeah', 'yesterday', 'yet', 'you', 'your', 'yourself']
256 47
Epoch [100/1000], Loss: 1.0307
Epoch [200/1000], Loss: 0.0130
Epoch [300/1000], Loss: 0.0035
Epoch [400/1000], Loss: 0.0005
Epoch [500/1000], Loss: 0.2116
Epoch [600/1000], Loss: 0.0001
Epoch [700/1000], Loss: 0.0000
Epoch [800/1000], Loss: 0.0000
Epoch [900/1000], Loss: 0.0000
Epoch [1000/1000], Loss: 0.0000
final loss: 0.0000
training complete. file saved to data.pth
(.venv)
RIT-OALAB-13+RIT@RIT-OALAB-13 MINGW64 ~/chatbot-deployment (main)
$ python chat.py
Let's chat! (type 'quit' to exit)
You: hi, how are doing
i'm doing great. what about you?
You: i'm pretty good. thanks for asking
everything's been good with you?
You: i've been great. what about you?
i'm actually in school right now.
You: what school do you to?
I do not understand...
You: what school do you go to?
i'm attending pcc right now.
You: do you like it thwre?
it's not bad. there are a lot of people there.
You: until then.
talk to you later.
You: quit
(.venv)
RIT-OALAB-13+RIT@RIT-OALAB-13 MINGW64 ~/chatbot-deployment (main)
$