**CREATE A CHATBOT IN PYTHON**

**Team Leader : Saravanakkumar D**

**Team Member 1 : Kumarakabilan P**

**Team Member 2 : Vijay I**

**Team Member 3 : Veerendran A**

**Team Member 4 : Saran R**

**PACKAGES INSTALLATION :**

!pip install transformers flask

Requirement already satisfied: transformers in d:\anaconda\lib\site-packages (4.32.1)

Requirement already satisfied: flask in d:\anaconda\lib\site-packages (2.2.2)

Requirement already satisfied: filelock in d:\anaconda\lib\site-packages (from transformers) (3.9.0)

Requirement already satisfied: huggingface-hub<1.0,>=0.15.1 in d:\anaconda\lib\site-packages (from transformers) (0.15.1)

Requirement already satisfied: numpy>=1.17 in d:\anaconda\lib\site-packages (from transformers) (1.24.3)

Requirement already satisfied: packaging>=20.0 in d:\anaconda\lib\site-packages (from transformers) (23.1)

Requirement already satisfied: pyyaml>=5.1 in d:\anaconda\lib\site-packages (from transformers) (6.0)

Requirement already satisfied: regex!=2019.12.17 in d:\anaconda\lib\site-packages (from transformers) (2022.7.9)

Requirement already satisfied: requests in d:\anaconda\lib\site-packages (from transformers) (2.31.0)

Requirement already satisfied: tokenizers!=0.11.3,<0.14,>=0.11.1 in d:\anaconda\lib\site-packages (from transformers) (0.13.2)

Requirement already satisfied: safetensors>=0.3.1 in d:\anaconda\lib\site-packages (from transformers) (0.3.2)

Requirement already satisfied: tqdm>=4.27 in d:\anaconda\lib\site-packages (from transformers) (4.65.0)

Requirement already satisfied: Werkzeug>=2.2.2 in d:\anaconda\lib\site-packages (from flask) (2.2.3)

Requirement already satisfied: Jinja2>=3.0 in d:\anaconda\lib\site-packages (from flask) (3.1.2)

Requirement already satisfied: itsdangerous>=2.0 in d:\anaconda\lib\site-packages (from flask) (2.0.1)

Requirement already satisfied: click>=8.0 in d:\anaconda\lib\site-packages (from flask) (8.0.4)

Requirement already satisfied: colorama in d:\anaconda\lib\site-packages (from click>=8.0->flask) (0.4.6)

Requirement already satisfied: fsspec in d:\anaconda\lib\site-packages (from huggingface-hub<1.0,>=0.15.1->transformers) (2023.4.0)

Requirement already satisfied: typing-extensions>=3.7.4.3 in d:\anaconda\lib\site-packages (from huggingface-hub<1.0,>=0.15.1->transformers) (4.7.1)

Requirement already satisfied: MarkupSafe>=2.0 in d:\anaconda\lib\site-packages (from Jinja2>=3.0->flask) (2.1.1)

Requirement already satisfied: charset-normalizer<4,>=2 in d:\anaconda\lib\site-packages (from requests->transformers) (2.0.4)

Requirement already satisfied: idna<4,>=2.5 in d:\anaconda\lib\site-packages (from requests->transformers) (3.4)

Requirement already satisfied: urllib3<3,>=1.21.1 in d:\anaconda\lib\site-packages (from requests->transformers) (1.26.16)

Requirement already satisfied: certifi>=2017.4.17 in d:\anaconda\lib\site-packages (from requests->transformers) (2023.7.22)

WARNING: Skipping D:\anaconda\Lib\site-packages\tensorflow\_io\_gcs\_filesystem-0.31.0.dist-info due to invalid metadata entry 'name'

WARNING: Skipping D:\anaconda\Lib\site-packages\tensorflow\_io\_gcs\_filesystem-0.31.0.dist-info due to invalid metadata entry 'name'

**Preprocessing :**

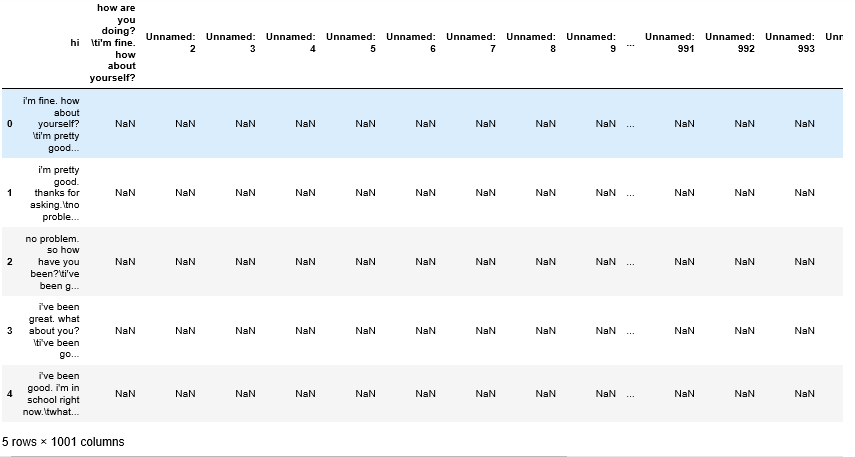
!pip install pandas

!pip install numpy

!pip install nltk

df = pd.read\_csv("dialogs.csv")

df.head()



df.info()

<class 'pandas.core.frame.DataFrame'> RangeIndex: 3724 entries, 0 to 3723 Columns: 1001 entries, hi to Unnamed: 1000 dtypes: float64(996), object(5) memory usage: 28.4+ MB

df = df.drop\_duplicates()

df = df.dropna()

df['message'] = df['message'].str.lower()

df['message'] = df['message'].str.replace('[^\w\s]', '')

nltk.download('punkt')

nltk.download('stopwords')

from nltk.corpus import stopwords

stop\_words = set(stopwords.words('english'))

from nltk.stem import WordNetLemmatizer

lemmatizer = WordNetLemmatizer()

df.to\_csv("cleaned\_dialogs.csv", index=False)

**Web app development :**

. . .

from transformers import GPT2LMHeadModel, GPT2Tokenizer

from flask import Flask, request, jsonify

​

app = Flask(\_\_name\_\_)

​

# Load the GPT-2 model and tokenizer

model\_name = "gpt2"

model = GPT2LMHeadModel.from\_pretrained(model\_name)

tokenizer = GPT2Tokenizer.from\_pretrained(model\_name)

​

@app.route("/chat", methods=["POST"])

def chat():

user\_input = request.json["user\_input"]

​

# Generate a response from the chatbot

input\_ids = tokenizer.encode(user\_input, return\_tensors="pt")

response\_ids = model.generate(input\_ids, max\_length=100, num\_return\_sequences=1, no\_repeat\_ngram\_size=2)

response = tokenizer.decode(response\_ids[0], skip\_special\_tokens=True)

​

return jsonify({"response": response})

​

if \_\_name\_\_ == "\_\_main\_\_":

app.run(debug=True)

​

. . .

user\_input = widgets.Text(placeholder="Type your message...")

output = widgets.Output()

​

def on\_send\_message(change):

if change.new and change.new[-1] == '\n':

with output:

clear\_output()

user\_message = user\_input.value.strip() # Remove leading/trailing whitespace

display(widgets.HTML("<strong>You:</strong> " + user\_message))

​

# Send user input to the chatbot and display the response

response = get\_chatbot\_response(user\_message)

display(widgets.HTML("<strong>Chatbot:</strong> " + response))

user\_input.value = "" # Clear the input field

​

user\_input.observe(on\_send\_message, names='value')

display(user\_input)

display(output)

output: Text(value='', placeholder='Type your message...')