

PHASE-4: DEVELOPMENT PART-2

PROJECT 3 – CREATE A CHATBOT USING PYTHON

In the previous phase of our project, we have cleaned our dataset and constructed a simple chatbot that responds very simply to user inputs. In this phase we will extend the capabilities of our chatbot by integrating a machine learning model.

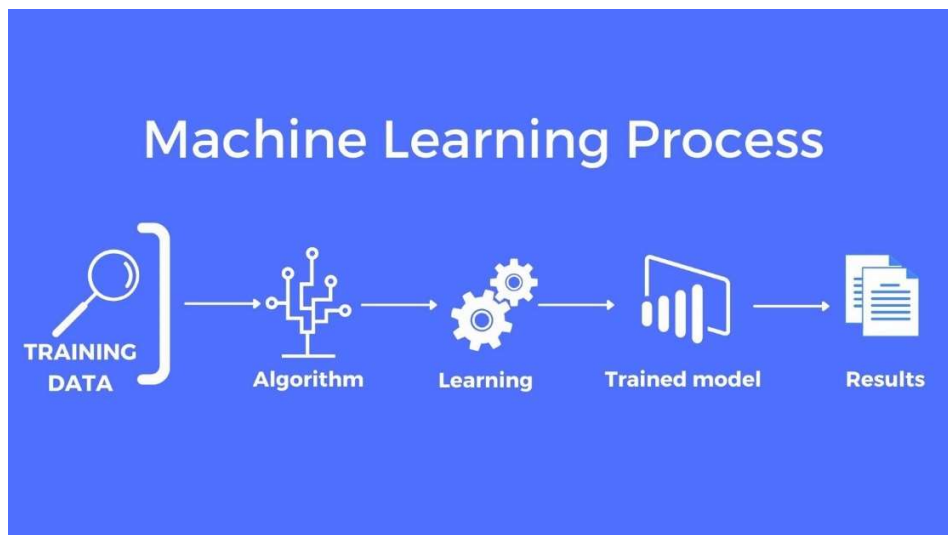
MACHINE LEARNING MODEL:

A machine learning model is a mathematical representation of a problem domain that, through training on data, acquires the ability to generalize and make predictions or classifications on new, unseen data. It learns from historical examples, identifying patterns and relationships within the data to perform tasks like prediction, classification, or decision-making.

Machine learning manages data by splitting it into training, validation, and test datasets.

- Training data is used to train the model by exposing it to a large set of examples.
- Validation data is used to fine-tune the model and optimize hyperparameters, ensuring it generalizes well.
- Test data is kept separate and used to evaluate the model's performance on unseen data, providing an unbiased assessment of its effectiveness.

This data management strategy ensures that the model learns from one set of data, validates on another, and is ultimately tested on a different dataset to assess its real-world applicability.



Our Helping Hand:

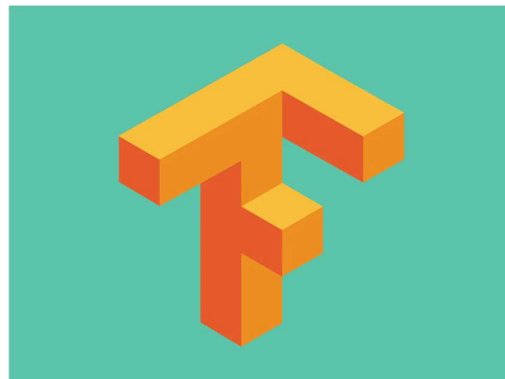
The machine learning model aids the chatbot by enabling it to understand and generate human-like responses. The model is trained on a dataset containing user inputs and corresponding chatbot responses. It learns the language patterns and context from this training data, allowing the chatbot to provide relevant and coherent replies to user queries. The model's ability to generalize from the training data helps the chatbot engage in dynamic and context-aware conversations, making it a more effective and responsive conversational agent.

Tensor Flow is our chosen ML framework, we are about to work with. Before diving in let's take a quick look at why tensor flow is an apt ML frame work for our purpose.

TENSOR FLOW:

TensorFlow is a popular open-source machine learning framework that is widely used for building and training deep learning models, including neural networks for natural language processing tasks like chatbots.

- TensorFlow provides a high-level API for building neural network models.
- TensorFlow includes built-in tools for training neural networks. For e.g., the `model.compile` and `model.fit` functions are used to define the training process, loss function, and optimizer.
- TensorFlow supports *GPU acceleration*, which can significantly speed up the training of deep learning models. This is crucial for training large chatbot models on extensive datasets



CODE SNIPPET:

```
import tensorflow as tf
import numpy as np
import pandas as pd

file_path="C:\\Users\\shiva\\OneDrive\\Desktop\\cleaned_unique_dialogs.csv"
                                                    #DATASET file
df = pd.read_csv(file_path)

# Extract user inputs and chat responses, and clean the text data
user_inputs = df['user_input'].astype(str).str.lower()
chat_responses = df['chat_response'].astype(str).str.lower()
```

```

# Tokenization
max_vocab_size = 10000
tokenizer = tf.keras.layers.TextVectorization(max_tokens=max_vocab_size)
tokenizer.adapt(user_inputs + chat_responses)
user_inputs = tokenizer(user_inputs)
chat_responses = tokenizer(chat_responses)

max_sequence_length = 10
user_inputs = tf.keras.preprocessing.sequence.pad_sequences(user_inputs,
maxlen=max_sequence_length, padding='post')
chat_responses = tf.keras.preprocessing.sequence.pad_sequences(chat_responses,
maxlen=max_sequence_length, padding='post')

# Define the model
model = tf.keras.Sequential([
    tf.keras.layers.Embedding(max_vocab_size, 64, mask_zero=True),
    tf.keras.layers.LSTM(128, return_sequences=True),
    tf.keras.layers.TimeDistributed(tf.keras.layers.Dense(max_vocab_size,
                                                              activation='softmax'))])

# Compile the model
model.compile(optimizer='adam', loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])

# Training the model
model.fit(user_inputs, chat_responses, epochs=50)

# Define a function to generate responses
def generate_response(input_text):
    input_text = [input_text]
    input_seq = tokenizer(input_text)
    predicted_seq = model.predict(input_seq)
    # Reverse the tokenization process
    predicted_text = " ".join([tokenizer.get_vocabulary()[index] for index in
                               np.argmax(predicted_seq, axis=-1)[0]])

    return predicted_text

while True:
    user_input = input("User: ")
    if user_input.lower() == "exit":
        print("Chatbot: Goodbye!")
        break # Exit the conversation loop
    response = generate_response(user_input)
    print("Chatbot: " + response)

```

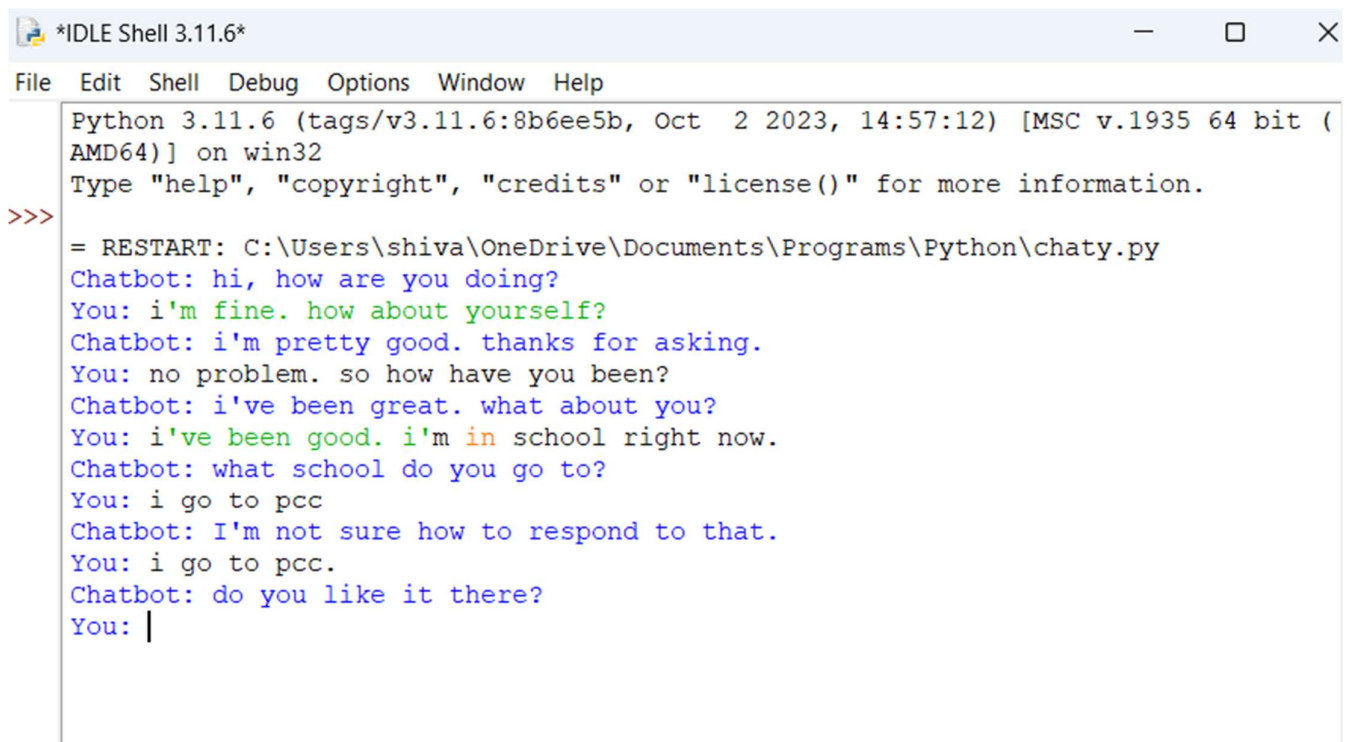
LIMITATIONS:

- This chatbot was trained using a very simple ML model with very less data to support.
- This chatbot may not capture the complexity of human language as well as more advanced models like transformer-based architectures. Real chatbots often use more sophisticated models.
- The model is trained for a fixed number of epochs (50 in this code), but in practice, training might require many more epochs to achieve good performance. Training a chatbot is an ongoing process.
- The code does not include fine-tuning or hyperparameter optimization, which is often required for achieving the best model performance.
- In practice, a chatbot would have a user interface for user interaction, which is not included in this code.

ACHIEVEMENTS:

- The code enables the chatbot to have natural and meaningful conversations with users, improving the user experience.
- The chatbot can understand and maintain context in conversations, providing relevant responses.
- It can be customized for different use cases by training on specific datasets, making it versatile.

OUTPUT:



```
*IDLE Shell 3.11.6*
File Edit Shell Debug Options Window Help
Python 3.11.6 (tags/v3.11.6:8b6ee5b, Oct 2 2023, 14:57:12) [MSC v.1935 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\shiva\OneDrive\Documents\Programs\Python\chaty.py
Chatbot: hi, how are you doing?
You: i'm fine. how about yourself?
Chatbot: i'm pretty good. thanks for asking.
You: no problem. so how have you been?
Chatbot: i've been great. what about you?
You: i've been good. i'm in school right now.
Chatbot: what school do you go to?
You: i go to pcc
Chatbot: I'm not sure how to respond to that.
You: i go to pcc.
Chatbot: do you like it there?
You: |
```