**CREATE A CHATBOT IN PYTHON**

**PHASE 3: DEVELOPMENT PART 1**

Building the dataset model by loading and pre-processing dataset.

Steps to follow when we begin to import the necessary libraries are:

1.Natural Language Processing (NLP) Libraries:

* NLTK (Natural Language Toolkit): NLTK is a comprehensive library for working with human language data. It provides tools for tokenization, stemming, tagging, parsing, and more.
* spaCy: spaCy is another popular NLP library that’s designed for production use. It’s known for its speed and efficiency in various NLP tasks.

2. Machine Learning Frameworks:

* scikit-learn: If your chatbot involves machine learning, scikit-learn is a versatile library for various machine learning tasks, including text classification and clustering.
* TensorFlow and Keras:These libraries are great for building deep learning models, including neural networks for more advanced chatbot applications.
* PyTorch: PyTorch is another deep learning framework that’s widely used for natural language processing tasks.

3. Text Processing Libraries:

* TextBlob:TextBlob simplifies text processing in Python. It provides easy-to-use methods for common NLP tasks like part-of-speech tagging, noun phrase extraction, and sentiment analysis.
* Gensim:Gensim is a library for topic modeling and document similarity analysis. It’s helpful if your chatbot needs to understand and generate text based on topics.

4. Chatbot Development Frameworks:

* ChatterBot:ChatterBot is a Python library that simplifies chatbot development by providing a straightforward way to create conversational agents using machine learning and NLP.
* Rasa:Rasa is an open-source chatbot framework that’s highly extensible and allows you to build chatbots with natural language understanding and dialogue management capabilities.

5. Web Frameworks :

-Flask or Django: If you want to create a web-based chatbot, Flask or Django can be used to build the web application to interact with users through a browser.

**#import the libraries**

python

Import nltk

From nltk.chat.util import Chat, reflections

Pairs = [

[‘my name is (.\*)’, [‘Hi %1! How can I assist you today?’]],

[‘(hi|hello|hey)’, [‘Hello there! How can I help you?’]],

[‘(.\*) (help|assist)’, [‘Sure, I can help. What do you need assistance with?’]],

[‘(.\*) (weather|temperature)’, [‘Sorry, I am not programmed to provide weather information.’]],

[‘(.\*)’, [‘I’m sorry, but I am unable to understand. Can you please rephrase?’]]

]

Chatbot = Chat(pairs, reflections)

Chatbot.converse()

**#load the dataset**

Df=pd.read\_csv(‘ /kaggle/input/simple-dialogs-for-chatbot/dialogs.txt’ ,sep=’\t’ ,names=[‘question’ , ‘answer’])

Printf(f’ Dataframe size: {len(df)}’)

Df.head()

**PROGRAM**

python

import tensorflow as tf

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Dense, Embedding, LSTM

model = Sequential()

model.add(Embedding(vocab\_size, embedding\_dim, input\_length=max\_seq\_length))

model.add(LSTM(units=128))

model.add(Dense(units=vocab\_size, activation='softmax'))

model.compile(optimizer='adam', loss='categorical\_crossentropy', metrics=['accuracy'])

model.fit(X\_train, y\_train, epochs=10, batch\_size=32)

response = model.predict\_classes(user\_input)

**CONCLUSION**

In this project, we have introduced a chatbot that is able to interact with users. This chatbot can answer queries in the textual user input. For this purpose, AIML with program-o has been used. The chatbot can answer only those questions which he has the answer in its AIML dataset.