TITILE: CREATE A CHATBOT USING PYTHON

NAME: HANISHKHA. K

PREPROCESSING THE DATASET

INTRODUCTION:

This document discusses the preprocessing steps to prepare the dataset for creating a chatbot in python. Dataset preprocessing is a technique of converting the raw data into clean dataset.

DATA SOURCES:

kaggle/input/simple-dialogs-for-chatbot/dialogs.txt

DATA PREPROCESSING

Preprocessing of the dataset involved several essential tasks:

- Manage a missing data
- Feature extraction
- Datatype conversion

1. Manage a missing data:

Missing values can be predicted by some techniques such as forward-fill or backward-fill to ensure a complete dataset. The goal of this process is to reformat the data into formatted data.

2. Feature extraction:

Additing of extra features were created to enhance the dataset's predictive power. This may include transformations, scaling, or the creation of new derived features. The process is used to smooth the large amount of data.

3. DATA TYPE CONVERSION:

Data types were used to consistency of our dataset. In particular non-numeric data types were converted to numerical types to make them compatible with machine learning algorithm.

PROGRAM:

```
import tensorflow as tf

import numpy as np

import pandas as pd

 import matplotlib.pyplot as plt

import seaborn as sns from tensorflow.keras.layers

import TextVectorization

import restring from tensorflow.keras.layers

import LSTM,Dense,Embedding,Dropout,LayerNormalization

df=pd.read_csv('/kaggle/input/simpledialogsforchatbot/dialogs.txt',sep='\t',na

mes=['question','answer'])

print(f'Dataframe size: {len(df)}') df.head()

df[‘question tokens’]=df[‘question’].apply(lambda x:len(x.split()))

df[‘answer tokens’]=df[‘answer’].apply(lambda x:len(x.split()))

plt.style.use(‘fivethirtyeight’)

fig,ax=plt.subplots(nrows=1,ncols=2,figsize=(20,5))

sns.set_palette(‘Set2’)

sns.histplot(x=df[‘question tokens’],data=df,kde=True,ax=ax[0])

sns.histplot(x=df[‘answer tokens’],data=df,kde=True,ax=ax[1])

sns.jointplot(x=’question tokens’,y=’answer

tokens’,data=df,kind=’kde’,fill=True,cmap=’YlGnBu’)

plt.show()

history=model.fit(

   train_data,

   epochs=100,
```

```
    validation_data=val_data,
    callbacks=[
        tf.keras.callbacks.TensorBoard(log_dir='logs'),
        tf.keras.callbacks.ModelCheckpoint('ckpt',verbose=1,save_best_only=True)
    ])
```

OUTPUT:

```
Epoch 1/100
23/23 [==============================] – ETA: 0s – loss: 1.6590 – accuracy: 0.2180
Epoch 1: val_loss improved from inf to 1.21875, saving model to ckpt
23/23 [==============================] – 68s 3s/step – loss: 1.6515 – accuracy:
0.2198 – val_loss: 1.2187 – val_accuracy: 0.3072
Epoch 2/100
23/23 [==============================] – ETA: 0s – loss: 1.2327 – accuracy: 0.3087
Epoch 2: val_loss improved from 1.21875 to 1.10877, saving model to ckpt
23/23 [==============================] – 53s 2s/step – loss: 1.2287 – accuracy:
0.3092 – val_loss: 1.1088 – val_accuracy: 0.3415
Epoch 3/100……
Epoch 100/100
23/23 [==============================] - ETA: 0s - loss: 0.3358 - accuracy: 0.6787
Epoch 100: val_loss did not improve from 0.33265
23/23 [==============================] - 22s 968ms/step - loss: 0.3385 - accuracy: 0.6773 - val_loss
: 0.3742 - val_accuracy: 0.6796
```

EXPLANATION FOR ABOVE PROGRAM:

Importing all necessary libraries and extract a data source to preprocess our data well good. By using a data source we will predict a text conversation. Using the concept of machine learning we tokenized the dataset.

Conclusion

In this above document we discuss about the preprocessing of the dataset to create a python chatbot and the steps involved in the data preprocessing. We will done our design with the help of this data source and build our plan.