IBM- Naan Mudhalvan Project phase -3

Student Name : Hemasree.B.M

Register Number : 620821243044

Branch : B.TECH AI&DS

Year : 3rd year

Topic : Artificial Intelligence

Title : To Create a Chatbot Using Python

College : Gnanamani College Of technology

**Load the Dataset:**

Load dataset into Python, which could be in formats like CSV, JSON, or plain text.

def load\_Dataset(data,size=None):

if(size!=None):

y,X=data[:size]

else:

y,X=data

X\_tokenizer=tokenize(X)

y\_tokenizer=tokenize(y)

X\_tensor=vectorization(X\_tokenizer,X)

y\_tensor=vectorization(y\_tokenizer,y)

return X\_tensor,X\_tokenizer, y\_tensor, y\_tokenizer

size=30000

data=preprocessed\_answers,preprocessed\_questions\

X\_tensor,X\_tokenizer, y\_tensor, y\_tokenizer=load\_Dataset(data,size)

# Calculate max\_length of the target tensors

max\_length\_y, max\_length\_X = y\_tensor.shape[1], X\_tensor.shape[1],

**Text Cleaning:**

Remove any irrelevant characters, symbols, or special characters.

Convert text to lowercase to ensure consistency.

**Tokenization:**

Split the text into individual words or tokens. This helps the model understand the structure of sentences.

**Stopword Removal:**

Remove common words (e.g., “the”, “and”, “is”) that don’t contribute much meaning to the text.

**Lemmatization or Stemming:**

Reduce words to their base or root form to normalize the text (e.g., “running” to “run”). You can choose either lemmatization or stemming based on your preference.

**Handling Abbreviations and Acronyms:**

Expand abbreviations and acronyms to their full forms for consistency and better understanding.

**Handling Contractions:**

Expand contractions (e.g., “can’t” to “cannot”) for uniformity in the text.

**Removing Duplicates:**

Eliminate duplicate entries to ensure data quality and avoid bias during training.

**Text Vectorization:**

Convert the text data into a numerical format suitable for model training. Common approaches include using bag-of-words, TF-IDF (Term Frequency-Inverse Document Frequency), or embeddings like Word2Vec or GloVe.

**Padding and Truncation:**

Ensure that all input sequences have a consistent length by either padding shorter sequences or truncating longer ones.

**Dataset Splitting:**

Divide the dataset into training, validation, and testing sets for model training and evaluation

.X\_train, X\_val, y\_train, y\_val = train\_test\_split(X\_tensor, y\_tensor, test\_size=0.2)

# Show length

print(len(X\_train), len(y\_train), len(X\_val), len(y\_val))

2980 2980 745 745

**Data preprocessing**

**The basic text processing in NLP are:**

1. Sentence Segmentation

2. Normalization

3. Tokenization

**1.Segmentation:**

In[1] :

import tensorflow as tf

from sklearn.model\_selection import train\_test\_split

#nlp processing

import unicodedata

import re

import numpy as np

import warnings

warnings.filterwarnings('ignore')

/opt/conda/lib/python3.10/site-packages/scipy/\_\_init\_\_.py:146: UserWarning: A NumPy version >=1.16.5 and <1.23.0 is required for this version of SciPy (detected version 1.23.5

warnings.warn(f"A NumPy version >={np\_minversion} and <{np\_maxversion}"

data=open('simple-dialogs-for-chatbot/dialogs.txt','r').read()

In[2] :

#paried list of question and corresponding answer

QA\_list=[QA.split('\t') for QA in data.split('\n')]

print(QA\_list[:5])

[['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."]]

In[3] :

questions=[row[0] for row in QA\_list]

answers=[row[1] for row in QA\_list]

In[4] :

print(questions[0:5])

print(answers[0:5])

['hi, how are you doing?', "i'm fine. how about yourself?", "i'm pretty good. thanks for asking.", 'no problem. so how have you been?', "i've been great. what about you?"]

["i'm fine. how about yourself?", "i'm pretty good. thanks for asking.", 'no problem. so how have you been?', "i've been great. what about you?", "i've been good. i'm in school right now."]

**2. Normalization:**

To reduce its randomness, bringing it closer to a predefined “standard”

In[5] :

def remove\_diacritic(text):

return ''.join(char for char in unicodedata.normalize('NFD',text)

if unicodedata.category(char) !='Mn')

In[6] :

def preprocessing(text):

#Case folding and removing extra whitespaces

text=remove\_diacritic(text.lower().strip())

#Ensuring punctuation marks to be treated as tokens

text=re.sub(r"([?.!,¿])", r" \1 ", text)

#Removing redundant spaces

text= re.sub(r'[" "]+', " ", text)

#Removing non alphabetic characters

text=re.sub(r"[^a-zA-Z?.!,¿]+", " ", text)

text=text.strip()

#Indicating the start and end of each sentence

text='<start> ' + text + ' <end>'

return text

preprocessed\_questions=[preprocessing(sen) for sen in questions]

preprocessed\_answers=[preprocessing(sen) for sen in answers]

print(preprocessed\_questions[0])

print(preprocessed\_answers[0])

<start> hi , how are you doing ? <end>

<start> i m fine . how about yourself ? <end>

**3.Tokenization:**

In[7] :

def tokenize(lang):

lang\_tokenizer = tf.keras.preprocessing.text.Tokenizer(

filters='')

#build vocabulary on unique words

lang\_tokenizer.fit\_on\_texts(lang)

return lang\_tokenizer

**Conclusion:**

In conclusion, the preprocessing phase is a crucial step in creating a chatbot using Python. It involves several key tasks, including text cleaning, tokenization, stemming or lemmatization, stop word removal, and handling special characters. This phase helps in preparing the text data for natural language processing tasks and ensures that the chatbot can understand and respond effectively to user input. Additionally, data preprocessing can also involve the collection and structuring of training data and the creation of a knowledge base or dataset to train the chatbot. By paying attention to the preprocessing phase, you can improve the accuracy and performance of your chatbot, making it a more efficient and user-friendly conversational agent.