CREATING CHATBOT USING PYTHON

TEAM MEMBER

911721104050:U. Karthick Raja

Phase-2 Innovation

Project:Creating Chatbot Using Python

OBJECTIVE:

The objective of this project is to create a chatbot in Python that provides exceptional customer service, answering user queries on a website or application. The objective is to deliver high-quality support to users, ensuring a positive user experience and customer satisfaction.

Phase 1: Problem Definition and Design Thinking

1.Design Thinking:

Define the scope of the chatbot's abilities, including: - Answering common questions related to diabetes. - Providing guidance on managing health and diabetes risk. - Directing users to appropriate resources for further information and support.

2.User Interface:

Determine integration points for the chatbot (website, app). - Design a user-friendly interface for seamless interactions with the chatbot

Natural Language Processing (NLP):

Implement NLP techniques to understand and process user input in a conversational manner.

PYTHON PROGRAM:

```
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')
#load the given datatset
data=open('E:\project\dialogs.txt','r').read()
#print Dataset values
head=[QA.split('\t') for QA in data.split('\n')]
print(“Dataset:”)
print(head[:5])
questions=[row[0] for row in QA_list]
answers=[row[1] for row in QA_list]
print(questions[0:5])
print(answers[0:5])
def remove_diacritic(word):
 return ''.join(char for char in unicodedata.normalize('NFD',text)
 if unicodedata.category(char) !='Mn')
def preprocessing(word):
 #Case folding and removing extra whitespaces
 word=remove_diacritic(word.lower().strip())
 #Ensuring punctuation marks to be treated as tokens
 word=re.sub(r"([?.!,¿])", r" \1 ", text)
 #Removing redundant spaces
word= re.sub(r'[" "]+', " ", text)
 #Removing non alphabetic characters
 word=re.sub(r"[^a-zA-Z?.!,¿]+", " ", text)
 word=word.strip()
```

```
 #Indicating the start and end of each sentence
 word='<start> ' + text + ' <end>'
 return word
#Tokenization
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
#Creating Dataset
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
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))
```

OUTPUT:

```
[['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."]]
```

Preporcessing:

```
<start> hi , how are you doing ? <end>
<start> i m fine . how about yourself ? <end>
```

4. Responses:

- ❒ **Plan responses for the chatbot, including: - Accurate answers to diabetes-related queries. - Suggestions for lifestyle changes. - Assistance in accessing medical resources.**

5. Integration:

- ❒ **Decide how the chatbot will be integrated with the website or app, ensuring a smooth user experience.**

6. Testing and Improvement:

- ❒ **Continuously test the chatbot's performance through real user interactions. - Gather user feedback and data to refine the chatbot's responses and capabilities.**

DATSOURCE :

Dataset Link: https://www.kaggle.com/datasets/grafstor/simple-dialogs-for-chatbot

CONCLUSION:

In Phase 1, We have to summarizes the problem, design thinking considerations, and dataset information for developing the AI-powered diabetes prediction system. It provides a structured framework for the initial phase of the project.