

AI - BASED CHATBOT USING PYTHON

BUILDING A RESPONSIVE CHATBOT

In Phase 3 of developing our chatbot, Building a chatbot in python involves several key steps, including loading and preprocessing the dataset, selecting relevant features, and preparing the data for model training. Here is a step-by-step process on how to begin this project



IMPORT LIBRARIES:

Importing libraries is a fundamental aspect of modern software development. Importing libraries allows developers to leverage the collective knowledge and expertise of the open-source community, enhancing the efficiency and robustness of their applications. Whether you're using Python's ``import`` statement libraries are a cornerstone of contemporary programming, empowering developers to create more powerful and feature-rich software.

```
#model
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')
```

DATA PREPROCESSING:

Designing a chatbot project involves several steps, and preprocessing the dataset is one of the critical initial steps. Preprocessing ensures that your data is clean, structured, and ready for analysis and model training. Here's a step-by-step guide on how to preprocess a dataset for a chatbot project, including various analyses you may need to perform

1. Data Collection:

Data collection for a chatbot refers to gathering, curating, and organizing the data needed to develop and train a chatbot. This data is crucial for enabling the chatbot to understand user input, generate appropriate responses, and continuously improve its performance. This could be text conversations, customer support logs, or any other relevant data.

dataset:

```
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.  
i've been good. i'm in school right now. what school do you go to?  
what school do you go to? i go to pcc.  
i go to pcc. do you like it there?  
do you like it there? it's okay. it's a really big campus.  
it's okay. it's a really big campus. good luck with school.  
good luck with school. thank you very much.  
how's it going? i'm doing well. how about you?  
i'm doing well. how about you? never better, thanks.  
never better, thanks. so how have you been lately?  
so how have you been lately? i've actually been pretty good. you?  
i've actually been pretty good. you? i'm actually in school right now.  
i'm actually in school right now. which school do you attend?  
which school do you attend? i'm attending pcc right now.  
i'm attending pcc right now. are you enjoying it there?  
are you enjoying it there? it's not bad. there are a lot of people there.  
it's not bad. there are a lot of people there. good luck with that.  
good luck with that. thanks.  
how are you doing today? i'm doing great. what about you?
```

SEGMENTATION:

Text segmentation, in the context of a chatbot or natural language processing (NLP), refers to the process of breaking down a continuous text or conversation into meaningful units or segments. These segments can help the chatbot better understand and respond to the user's input

```
#reading data  
data=open('/input/simple-dialogs-for-chatbot/dialogs.txt','r').read()  
  
#paried list of question and corresponding answer  
QA_list=[QA.split('\t') for QA in data.split('\n')]  
print(QA_list[:5])
```

OUTPUT:

```
[['hi, how are you doing?', 'i'm fine. how about yourself?'], ['i'm fine. how about yourself?', 'i'm p  
retty 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.']]
```

NORMALIZATION:

Text normalization in the context of a chatbot refers to the process of standardizing and transforming text input to make it more consistent and easier for the chatbot to process and understand. It involves various techniques aimed at cleaning and structuring text data.

```
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>
```

TOKENIZATION:

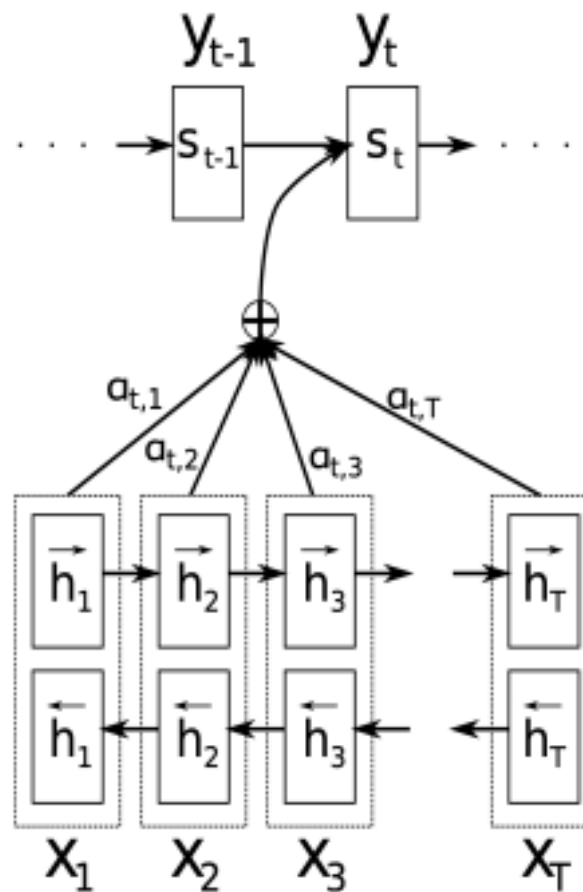
Tokenization is a crucial natural language processing (NLP) technique used in chatbots and various other text-processing applications. It involves the process of breaking down a continuous text string, typically a sentence or paragraph, into individual units known as "tokens." Tokens are usually words or subwords, which are the basic building blocks of text that the chatbot can then analyze, manipulate, or respond to.

```
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 :

In the context of a chatbot, a "dataset" refers to a structured collection of text or data that is used for various purposes, primarily for training and fine-tuning the chatbot's natural language processing (NLP) capabilities. Chatbot datasets typically consist of pairs or sequences of user messages and corresponding bot responses.

```
def vectorization(lang_tokenizer, lang):  
  
    #word embedding for training the neural network  
    tensor = lang_tokenizer.texts_to_sequences(lang)  
  
    tensor = tf.keras.preprocessing.sequence.pad_sequences(tensor,  
                                                            padding='post')  
  
    return tensor
```



```
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
```

TENSOR FLOW DATASET:

```
BUFFER_SIZE = len(X_train)
BATCH_SIZE = 64
steps_per_epoch = len(X_train)//BATCH_SIZE
embedding_dim = 256
units = 1024
vocab_inp_size = len(X_tokenizer.word_index)+1
vocab_tar_size = len(y_tokenizer.word_index)+1

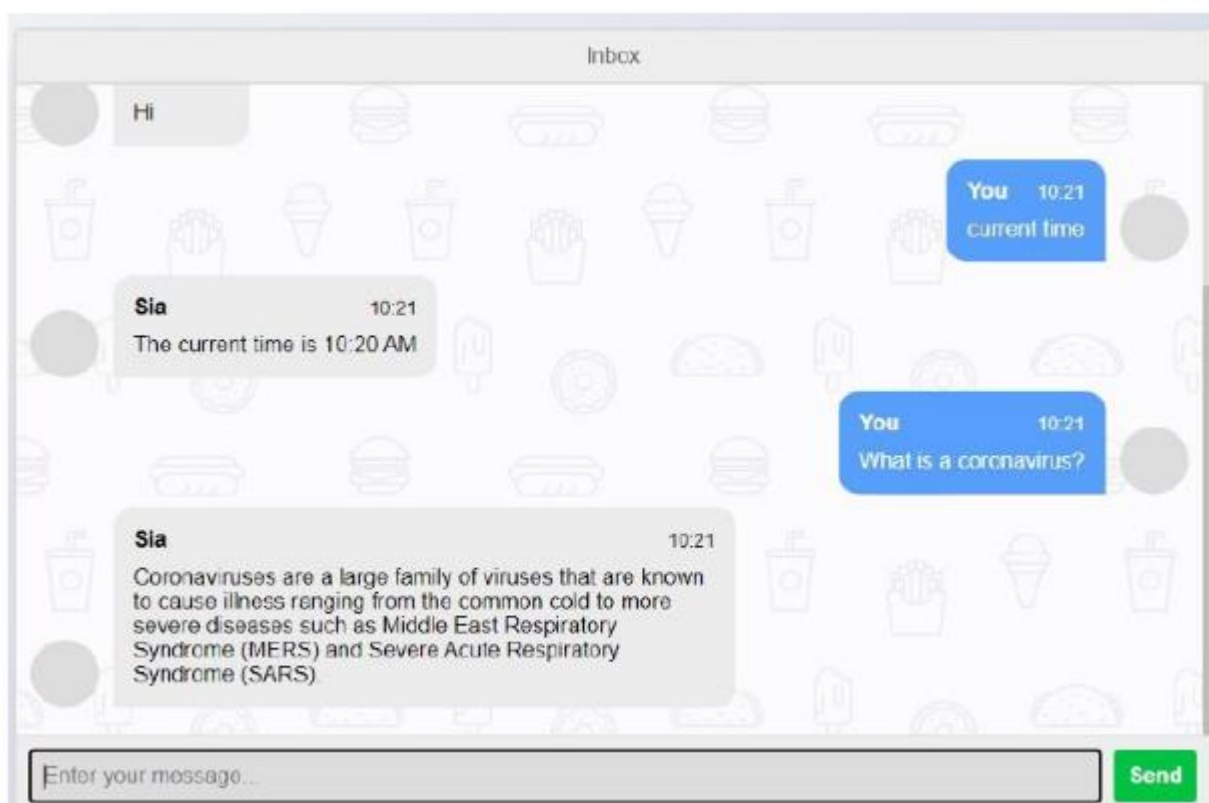
dataset = tf.data.Dataset.from_tensor_slices((X_train, y_train)).shuffle(BUFFER_SIZE)
dataset = dataset.batch(BATCH_SIZE, drop_remainder=True)

example_input_batch, example_target_batch = next(iter(dataset))
example_input_batch.shape, example_target_batch.shape
```

OUTPUT:

```
(TensorShape([64, 24]), TensorShape([64, 24]))
```

VISUALIZATION:



DESCRIPTION VISUALIZATION:

Text-based general chatbots have emerged as powerful tools, transforming the way we interact with technology and businesses. These AI-driven conversational agents have redefined customer service, marketing, and user engagement across various industries. With their text-based interface, they offer a seamless and user-friendly experience, ensuring accessibility to a wide audience.

These chatbots excel in quickly resolving customer inquiries, automating responses to frequently asked questions, and providing instant solutions. They enhance efficiency by freeing up human agents for more complex tasks.

Chatbots generate valuable data insights from user interactions. This data can be harnessed to understand user behavior, preferences, and pain points, aiding in strategic decision-making and marketing efforts.

In conclusion, text-based general chatbots have revolutionized the way businesses engage with their customers and manage inquiries. Their efficiency, scalability, and data-driven insights are reshaping the future of customer service and marketing, ensuring an enhanced user experience and improved operational effectiveness.

CONCLUSION:

By the end of Phase 3, we had successfully loaded and preprocessed our dataset, chosen relevant features, and defined our target variable. We are now well-prepared to move on to the subsequent phases, which involve Segmentation, Tokenization, Normalization training, and evaluation. Our ultimate goal is to develop an accurate text-based chatbot to benefit users. This phase is a crucial foundation for the rest of the project, as the quality of data and feature selection significantly impact the performance of the text-based chatbot model.