CREATE A CHATBOT IN PYTHON

Team Leader:

962821106033 : JERRIK PRAYWIN RAJ J

Team Members:

962821106004 : ABINANDH R S

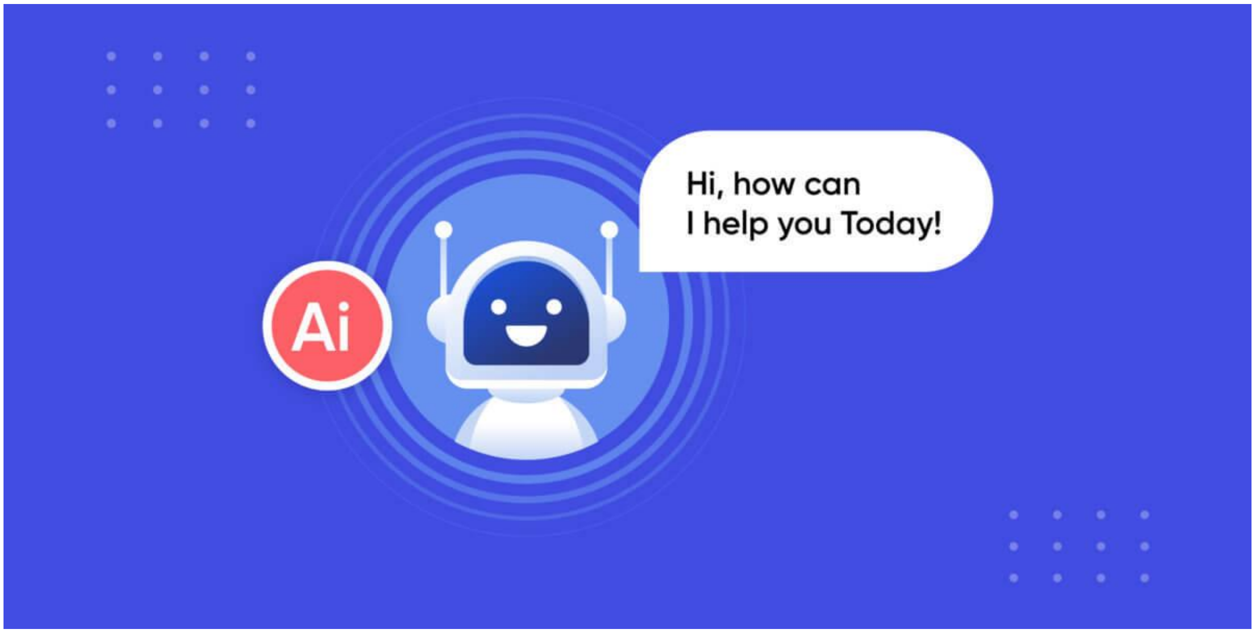
962821106014 : ARSHATH SHARIEF S

962821106017 : BALASUBRAMANIAN K

962821106018 : BASKARAN K

Phase-1 Document Submission

Project: *Creating a Chatbot*



OBJECTIVE:

The objective of this project is to create a chatbot using 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.Functionality

The scope of chatbots is continually expanding as technology advances and their capabilities grow. Chatbots have already found applications in various industries, and their potential continues to evolve.

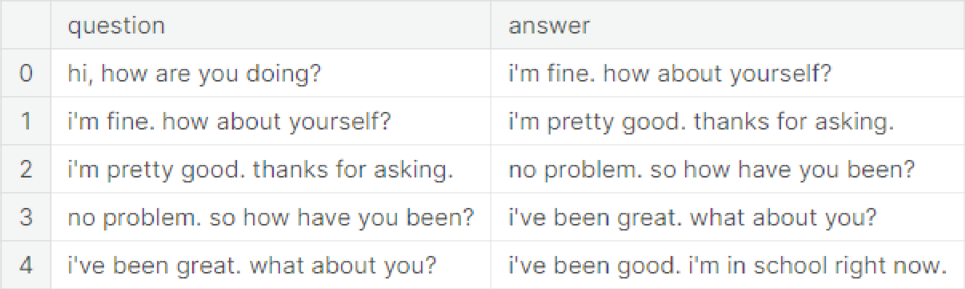
We are going to create a chatbot that answering common questions, providing guidance, and directing users to appropriate resources.

Program:

```
#import libraries

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 re,string
from tensorflow.keras.layers

import LSTM,Dense,Embedding,Dropout,LayerNormalization

df=pd.read_csv('/kaggle/input/simple-dialogs-for-

chatbot/dialogs.txt',sep='\t',names=['question','answer'])
print(f'Dataframe size: {len(df)}')
df.head()
```

Output:

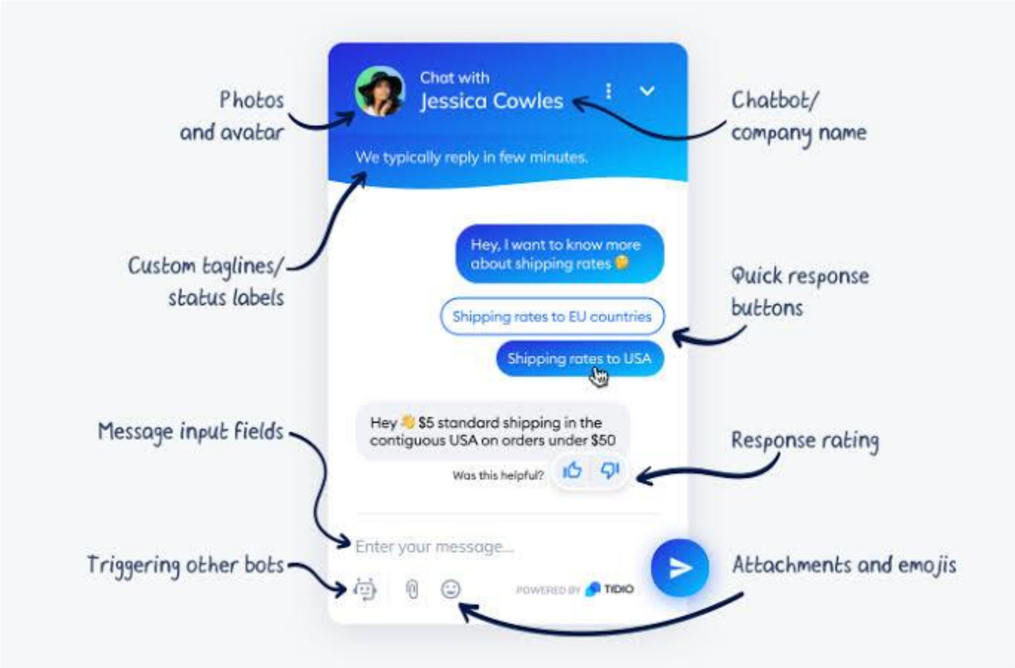
Dataframe size: 3725



	question	answer
0	hi, how are you doing?	i'm fine. how about yourself?
1	i'm fine. how about yourself?	i'm pretty good. thanks for asking.
2	i'm pretty good. thanks for asking.	no problem. so how have you been?
3	no problem. so how have you been?	i've been great. what about you?
4	i've been great. what about you?	i've been good. i'm in school right now.

2. User Interface:

We are planning to integrate our chatbot in both website and app because the existing chatbots are available in both app and website. So, we are planning to create our chatbot to keep up with the trend. The interface our chatbot will look like the below image (it may change or improve in future).



Chat with
Jessica Cowles

We typically reply in few minutes.

Hey, I want to know more about shipping rates 😊

Hey 👋 $5 standard shipping in the contiguous USA on orders under $50

Was this helpful?

Enter your message...

3. Natural Language Processing (NLP):

Natural Language Processing (NLP) is a subfield of artificial intelligence (AI). It helps machines process and understand the human language so that they can automatically perform repetitive tasks.

The basic text processing in NLP are:

1. Sentence Segmentation
2. Normalization
3. Tokenization

Sentence Segmentation:

The process of deciding from where the sentences actually start or end in NLP or we can simply say that here we are dividing a paragraph based on sentences. This process is known as Sentence Segmentation.

Program:

formatting data to be in a question answer format

```
#reading data
data=open('/kaggle/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])
```

```
[['hi, how are you doing?', "i'm fine. how about yourself?"], ["i'm fine. how abo
ut 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."]]
```

```
questions=[row[0] for row in QA_list]
answers=[row[1] for row in QA_list]


print(questions[0:5])
print(answers[0:5])
```

```
['hi, how are you doing?', "i'm fine. how about yourself?", "i'm pretty good. tha
nks for asking.", 'no problem. so how have you been?', "i've been great. what abo
ut you?"]
["i'm fine. how about yourself?", "i'm pretty good. thanks for asking.", 'no prob
lem. so how have you been?', "i've been great. what about you?", "i've been good.
i'm in school right now."]
```

Normalization:

Text normalization is a pre-processing step aimed at improving the quality of the text and making it suitable for machines to process. Four main steps in text normalization are case normalization, tokenization and stop word removal, Parts-of-Speech (POS) tagging, and stemming.

Program:

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

```python
def remove_diacritic(text):
    return ''.join(char for char in unicodedata.normalize('NFD',text)
                   if unicodedata.category(char) !='Mn')


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 used in natural language processing to split paragraphs and sentences into smaller units that can be more easily assigned mean ing. The first step of the NLP process is gathering the data (a sentence) and breaking it into understandable parts (words).

Program:

```
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
```

4.Dataset:

We are using the dataset provided in the problem. They provide some basic dialogues in question and answer which are commonly used conversation in day-to-day life.

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

5.Response:

Since, our chatbot is a basic model now it is only simple task and conversation. But it can give accurate answers, suggestion and assistance.

We will further develop our chatbot in accordance with user need.

6.Integration:

Choose a Chatbot Platform: Before integration, we need to select a chatbot development platform or framework. We are going to use popular options include Dialogflow, Microsoft Bot Framework, IBM Watson Assistant, Rasa, and custom-built solutions using programming languages like Python, JavaScript, or Ruby.

Develop the Chatbot: we will create and configure our chatbot on the chosen platform.

Obtain API Keys or Credentials: Most chatbot platforms provide API keys or credentials that allow our website or app to communicate with the chatbot. We will use these credentials for integration.

Embed the Chatbot Widget:

Website Integration:

Widget or Embed Code: Most chatbot platforms provide a widget or embed code that we can copy and paste into your website's HTML. This code typically includes a chat window or icon that users can click to open the chatbot.

App Integration:

In-App Chat Interface: Integrate the chatbot's interface directly into our app's user interface, allowing users to access the chatbot's functionalities within the app. This may require custom app development.

API Integration: Use the chatbot's API to send and receive messages from within your app. This method offers more flexibility but requires more custom development work.

Configure Webhooks (Optional): Depending on our chatbot's functionalities, we may need to set up webhooks to handle specific actions or external API integrations. For example, if our chatbot needs to fetch real-time data from an external source, we would configure webhooks to manage these requests and responses.

7.Testing and Improvement:

Test and Debug: We will thoroughly test the integrated chatbot to ensure it functions as expected. We will check for any issues with the chatbot's interaction flow, responsiveness, and data handling.

Monitor and Refine: After integration, we will monitor the chatbot's performance and gather user feedback. We will make adjustments and refinements to improve the chatbot's responses and user experience over time.

Security and Compliance: We will ensure that the integrated chatbot complies with security and privacy standards, especially if it handles sensitive user data. We will implement measures to protect user information and maintain data security.

Scalability: As our website or app grows, we will consider the scalability of our chatbot. We will ensure that it can handle increased user interactions and traffic without performance issues.

Updates and Maintenance: We will keep our chatbot and integration up-to-date. Updates may be required to accommodate changes in our website or app's design or functionality, as well as to improve the chatbot's capabilities.

Integrating a chatbot with a website or app is a dynamic process that requires coordination between developers, designers, and content creators. Our goal is to create a seamless and user-friendly experience that enhances the overall functionality and usability of our digital platform.

CONCLUSION:

Our chatbot project represents a significant step forward in leveraging AI and natural language processing to enhance user experiences. As we move into the future, we are excited to see how our chatbot continues to make a positive impact and look forward to the possibilities it holds for further innovation and growth.