Title:Chatbot in python

Import Libraries

- The code starts by importing the necessary libraries.
- The code then creates a variable called tf which is used to represent the tensorflow library.
- Next, it imports numpy and pandas libraries as well as matplotlib for plotting purposes.
- Then it imports seaborn library for data visualization purposes.
- Next, there are two lines of code that create a text vectorizer object in order to transform text into numerical values so that they can be processed with machine learning algorithms like neural networks or support vector machines (SVM).
- The code takes in a list of text file and creates one file with the contents of all the text files.

```
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
```

Dataset load

- The code starts by importing the pandas library.
- It then creates a Dataframe object called df that is initialized with the contents of archive.zip, which contains two columns: question and answer.
- The code then prints out the size of df to show how many rows it has in it.
- Next, head() is used on df to print out just the first few rows from df so we can see what they are without having to scroll through all of them.
- The next line uses read_csv() function on a file called /content/archive.zip that was downloaded from Google Drive using pd (Python data) package and reads in each row as an individual string separated by a comma into an array named 'answer'.
- The code reads a zip file and prints the size of the dataframe.

```
df=pd.read_csv('/content/archive.zip',sep='\t',names=['question','answe
r'])
print(f'Dataframe size: {len(df)}')
df.head()
```

```
Dataframe size: 3725
```

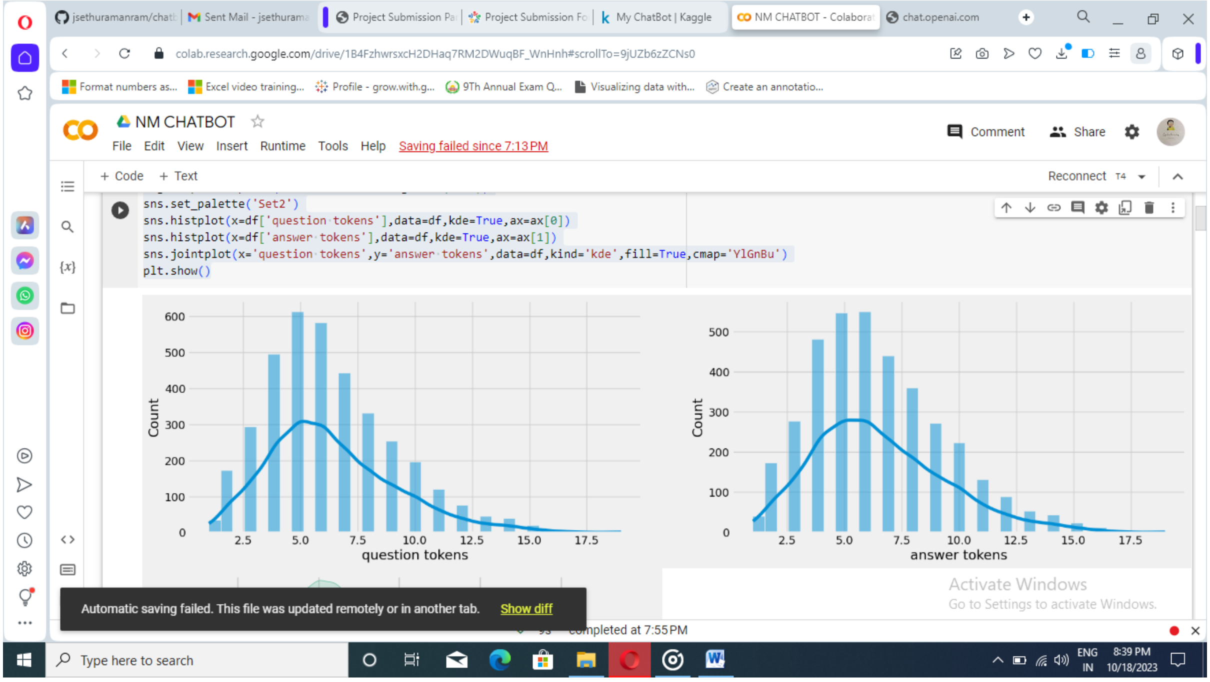
	question	answer
0	hi, how are you doing?	i'm fine. how about yourself?
1	i'm fine. how about yourself?	i'm pretty good. thanks for asking.
2	i'm pretty good. thanks for asking.	no problem. so how have you been?
3	no problem. so how have you been?	i've been great. what about you?
4	i've been great. what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

Data Preprocessing

- Next, the code sets up a figure with two subplots to display question tokens and answer tokens on different axes.
- The next line uses sns to create histograms of these data points using different colors for each axis (Set2).
- The next line plots both question tokens and answer tokens on their respective axes using jointplot().
- This is done so that we can see how many times each token appears in each set of data.
- The code is meant to show the distribution of tokens in a dataset.

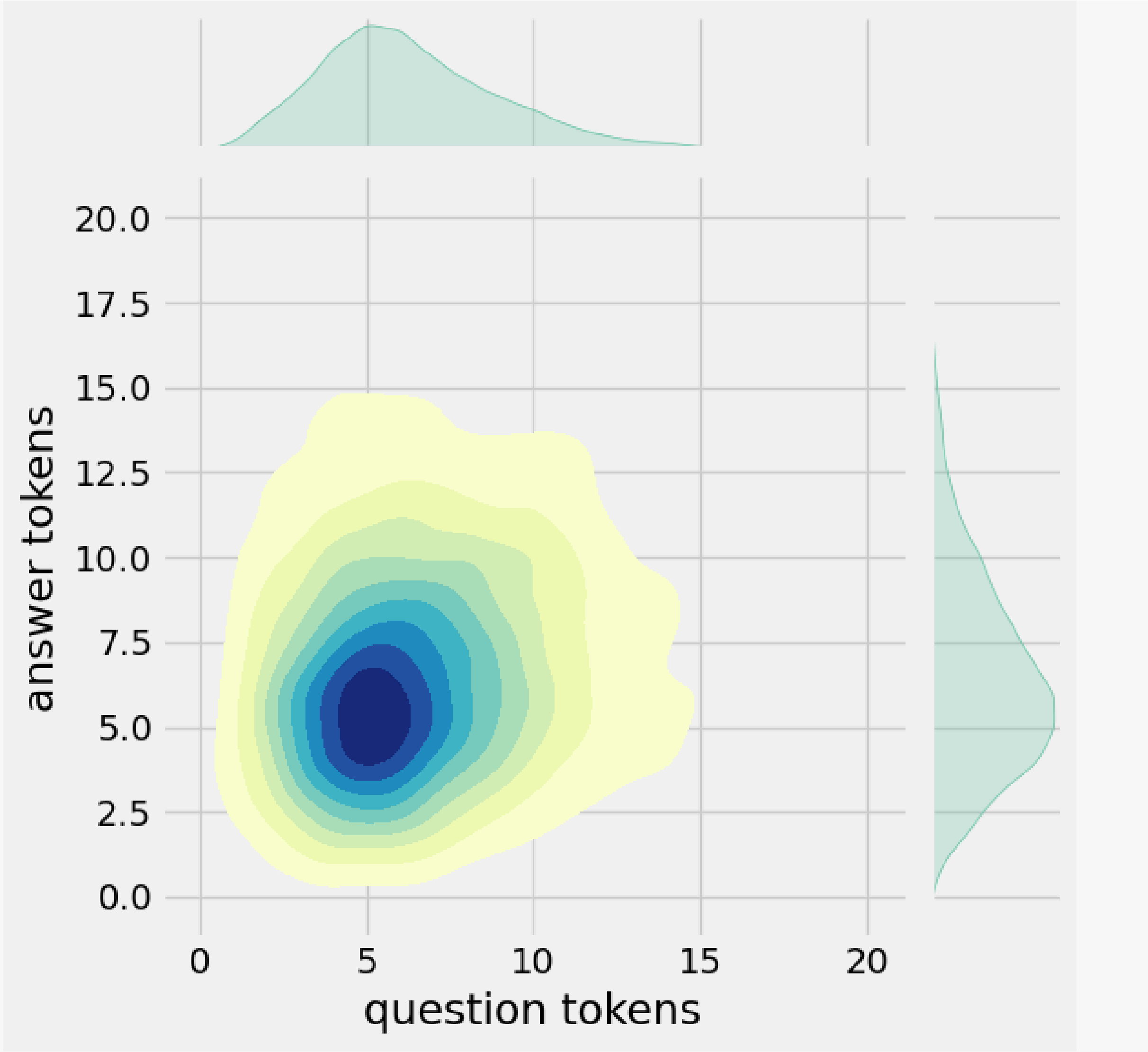
Data Visualization

```
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()
```



```
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()
```

Automatic saving failed. This file was updated remotely or in another tab.



Text Cleaning

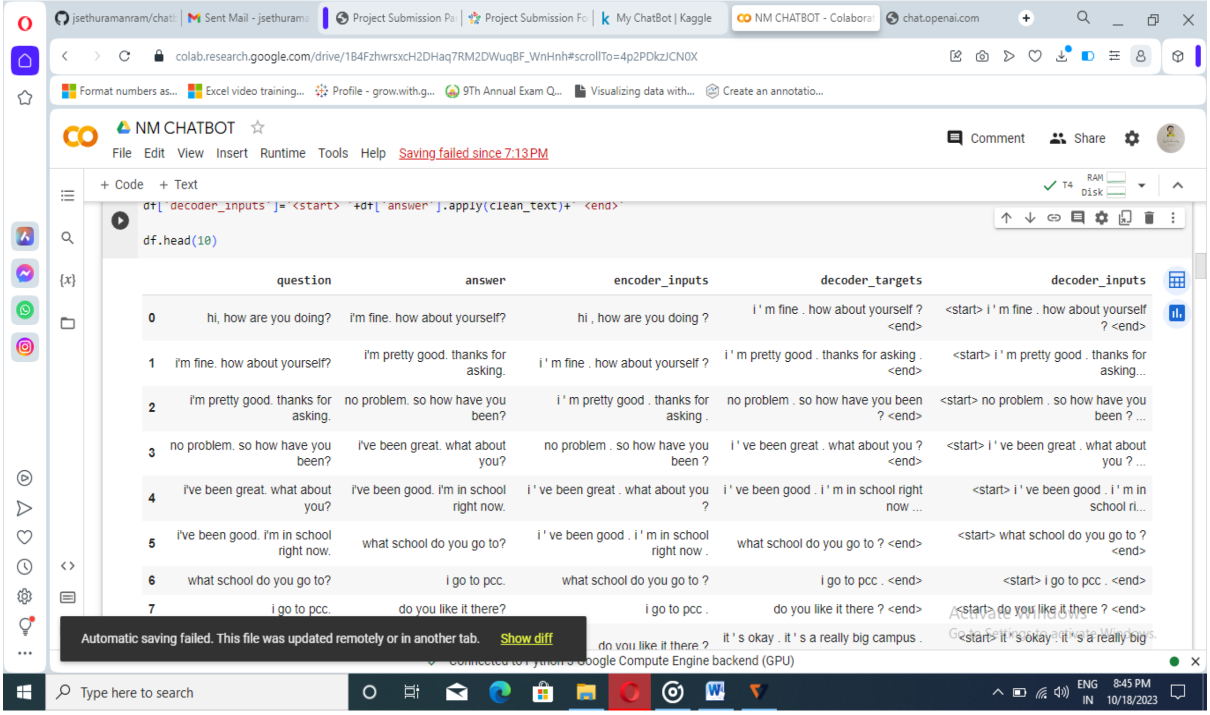
- DataFrame() which will create a new dataframe object for us to work on.
- Next we use df['question']=df['question'].apply(clean_text) which means that we are going to take all of our questions (in this case just one), clean them up so they're not full of punctuation or anything else that might be considered noise, then apply what's called string formatting on them so they'll look nice when printed out later on (e.g., "1" instead of 1).
- We do this because we want to make sure our encoding process doesn't mess up any words or sentences while trying to encode them into binary digits (0s and 1s).
- After doing this for every word in our list, we
- The code will take the text in column 1 of the dataframe and convert it to lowercase.
- It then takes the text in column 2, converts it to uppercase.
- It then converts all characters that are not letters into a single character, for example '1' becomes '1'.

```python
def clean_text(text):

    text=re.sub('-',' ',text.lower())
    text=re.sub('[.]',' . ',text)
    text=re.sub('[1]',' 1 ',text)
    text=re.sub('[2]',' 2 ',text)
    text=re.sub('[3]',' 3 ',text)
    text=re.sub('[4]',' 4 ',text)
    text=re.sub('[5]',' 5 ',text)
    text=re.sub('[6]',' 6 ',text)
    text=re.sub('[7]',' 7 ',text)
    text=re.sub('[8]',' 8 ',text)
    text=re.sub('[9]',' 9 ',text)
    text=re.sub('[0]',' 0 ',text)
    text=re.sub('[,]',' , ',text)
    text=re.sub('[?]',' ? ',text)
    text=re.sub('[!]',' ! ',text)
    text=re.sub('[$]',' $ ',text)
    text=re.sub('[&]',' & ',text)
    text=re.sub('[/]',' / ',text)
    text=re.sub('[:]',' : ',text)
    text=re.sub('[;]',' ; ',text)
    text=re.sub('[*]',' * ',text)
    text=re.sub('[\']',' \' ',text)
    text=re.sub('[\"]',' \" ',text)
    text=re.sub('\t',' ',text)
    return text

df.drop(columns=['answer tokens','question
tokens'],axis=1,inplace=True)
df['encoder_inputs']=df['question'].apply(clean_text)
df['decoder_targets']=df['answer'].apply(clean_text)+' <end>'
df['decoder_inputs']='<start> '+df['answer'].apply(clean_text)+' <end>'

df.head(10)
```



```
df['decoder_inputs']='<start> '+df['answer'].apply(clean_text)+' <end>'

df.head(10)
```

	question	answer	encoder_inputs	decoder_targets	decoder_inputs
0	hi, how are you doing?	i'm fine. how about yourself?	hi , how are you doing ?	i ' m fine . how about yourself ? <end>	<start> i ' m fine . how about yourself ? <end>
1	i'm fine. how about yourself?	i'm pretty good. thanks for asking.	i ' m fine . how about yourself ?	i ' m pretty good . thanks for asking . <end>	<start> i ' m pretty good . thanks for asking...
2	i'm pretty good. thanks for asking.	no problem. so how have you been?	i ' m pretty good . thanks for asking .	no problem . so how have you been ? <end>	<start> no problem . so how have you been ? ...
3	no problem. so how have you been?	i've been great. what about you?	no problem . so how have you been ?	i ' ve been great . what about you ? <end>	<start> 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.	i ' ve been great . what about you ?	i ' ve been good . i ' m in school right now ...	<start> i ' ve been good . i ' m in school ri...
5	i've been good. i'm in school right now.	what school do you go to?	i ' ve been good . i ' m in school right now .	what school do you go to ? <end>	<start> what school do you go to ? <end>
6	what school do you go to?	i go to pcc.	what school do you go to ?	i go to pcc . <end>	<start> i go to pcc . <end>
7	i go to pcc.	do you like it there?	i go to pcc .	do you like it there ? <end>	<start> do you like it there ? <end>
			do you like it there ?	it ' s okay . it ' s a really big campus .	<start> it ' s okay . it ' s a really big

Automatic saving failed. This file was updated remotely or in another tab. Show diff

Connected to Python 3 Google Compute Engine backend (GPU)