Social Media Comments Correction Using RNN's

1.Problem Statement, Data and Metric Analysis

A.Problem Statement :

The Aim of this case study is to build a Good RNN model that corrects the Noised, Non-Grammatical and mispelled social media comments to a better addressed, grammatically strong and rule based english coded sentences.

As we know while using social media, We have seen bunch of half-short forms of english words and sentences. So, to use RNN in state-of-the art Natural Language Processsing for Translation of sentences, we have tried approach.

This case study is based on Research Paper: Sentence Correction using Recurrent Neural Networks by:

Gene Lewis,Department of Computer Science

Stanford University Stanford

CA 94305

Link: http://cs224d.stanford.edu/reports/Lewis.pdf

B.How Deep Learning will help solving this problem?

As we have seen Sequence to Sequence translation of texts/sentences to predict possible words, Taking an example of search engine, You have seen possible suggestions over engine when you type some set of words.

Hence, to use that kind of Deep learning technique to build a model,to correctly present or predict the correct translation of sentences, we have headed off with this research paper.

This paper describes how character level and word level embeddings can be used as our inputs and thus, predicting right asnwers.

There are 2 parts for this, Encoder and Decoder models. That we will see in the future approach.

C.Data Definition :

Data is taken from English corpus of 2000 texts from the National University of Singapore https://www.comp.nus.edu.sg/~nlp/corpora.html.

NUS Social Media Text Normalization and Translation Corpus:

The corpus is created for social media text normalization and translation. It is built by randomly selecting 2,000 messages from the NUS English SMS corpus. The messages were first normalized into formal English and then translated into formal Chinese.

Example data from the testing set:

Input: "Ic...Haiz,nv ask me along?Hee,im so sian at hm."

Output: "I see. Sigh, why do you never ask me along? I’m so bored at home."

D.Metric Used :

There are three types of Metrics that is used to check our text similarities:

1. **Word and Character Based Perplexities**
2. **Word and Character Based Bleu Score**
3. **NIST Score**

2.Exploratory Data Analysis

A. Loading Libraries

In [ ]:

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

In [ ]:

```
import pandas as pd
%matplotlib inline
import warnings
warnings.filterwarnings("ignore")

import sqlite3
import pandas as pd
import numpy as np
import nltk
from keras.preprocessing.sequence import pad_sequences
import string
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.feature_extraction.text import TfidfTransformer
from sklearn.feature_extraction.text import TfidfVectorizer

from sklearn.feature_extraction.text import CountVectorizer
from sklearn.metrics import confusion_matrix
from sklearn import metrics
from sklearn.metrics import roc_curve, auc
from nltk.stem.porter import PorterStemmer
import re
import string
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer
from nltk.stem.wordnet import WordNetLemmatizer
import tensorflow as tf
# tf.compat.v1.enable_eager_execution()
from tensorflow import keras
from tensorflow.keras.layers import *
from tensorflow.keras.preprocessing import image
from tensorflow.keras.models import Model, load_model
from tensorflow.keras.layers import UpSampling2D
from tensorflow.keras.layers import MaxPooling2D, GlobalAveragePooling2D
from tensorflow.keras.layers import concatenate
from tensorflow.keras.layers import Multiply
from tensorflow.keras.callbacks import EarlyStopping, ModelCheckpoint
from tensorflow.keras import backend as K
from tensorflow.keras.layers import Input, Add, Dense, Activation, ZeroPadding2D, BatchNo
rmalization, Flatten, Conv2D, AveragePooling2D, MaxPooling2D, GlobalMaxPooling2D
from tensorflow.keras.models import Model, load_model
from tensorflow.keras.utils import plot_model
from tensorflow.keras.initializers import glorot_uniform
K.set_image_data_format('channels_last')
K.set_learning_phase(1)
import pickle

from tqdm import tqdm
import os
import tensorflow as tf
import plotly.offline as offline
import plotly.graph_objs as go
offline.init_notebook_mode()
from collections import Counter
```

B. Loading Data

In [ ]:

```
my_file = open("/content/drive/MyDrive/CASE STUDY 2/en2cn-2k.txt", "r")
content = my_file.read()

content_list = content.split("\n")
my_file.close()
```

In [ ]:

```
for i in content_list:
  if len(i)==0:
    content_list.remove(i)
```

In [ ]:

```
corr=[]
eng=[]

for i in range(len(content_list)):
  if i%3==0:
    corr.append(content_list[i])
  elif i%3==1:
    eng.append(content_list[i])
```

In [ ]:

```
final_df=pd.DataFrame()
final_df['Corrupted']=corr
final_df['English']=eng
```

In [ ]:

```
final_df
```

In [ ]:

```
import pickle
pickle.dump(final_df, open("/content/drive/MyDrive/CASE STUDY 2/final.pkl","wb"))
```

C. Analysis

In [ ]:

In [ ]:

```
import pickle
final=pickle.load(open("final.pkl", "rb"))
final.head()
```

Out[ ]:

	Corrupted	English
0	U wan me to "chop" seat 4 u nt?	Do you want me to reserve seat for you or not?
1	Yup. U reaching. We order some durian pastry a...	Yeap. You reaching? We ordered some Durian pas...
2	They become more ex oredi... Mine is like 25.....	They become more expensive already. Mine is li...
3	I'm thai. what do u do?	I'm Thai. What do you do?
4	Hi! How did your week go? Haven heard from you...	Hi! How did your week go? Haven't heard from y...

In [ ]:

```
final.columns
#There are onl 2 columns needed for our task,
#one is corrupted text and second one is corrected english texts respectively
```

Out[ ]:

```
Index(['Corrupted', 'English'], dtype='object')
```

In [ ]:

```
final.describe()

#There are 2000 unique words without any null values
```

Out[ ]:

	Corrupted	English
count	2000	2000
unique	2000	1989
top	Joey: Neo where u fr?	Ok.
freq	1	5

LENGTH OF CORRUPTED TEXTS AT CHARACTER LEVEL

In [ ]:

```
cor_len=[]
eng_len=[]

for i in final.values:
  cor_len.append(len(str(i[0])))
  eng_len.append(len(str(i[1])))

final['cor_len']=cor_len #storing the corrupted text character wise
final['eng_len']=eng_len #storing the english text character wise
```

In [ ]:

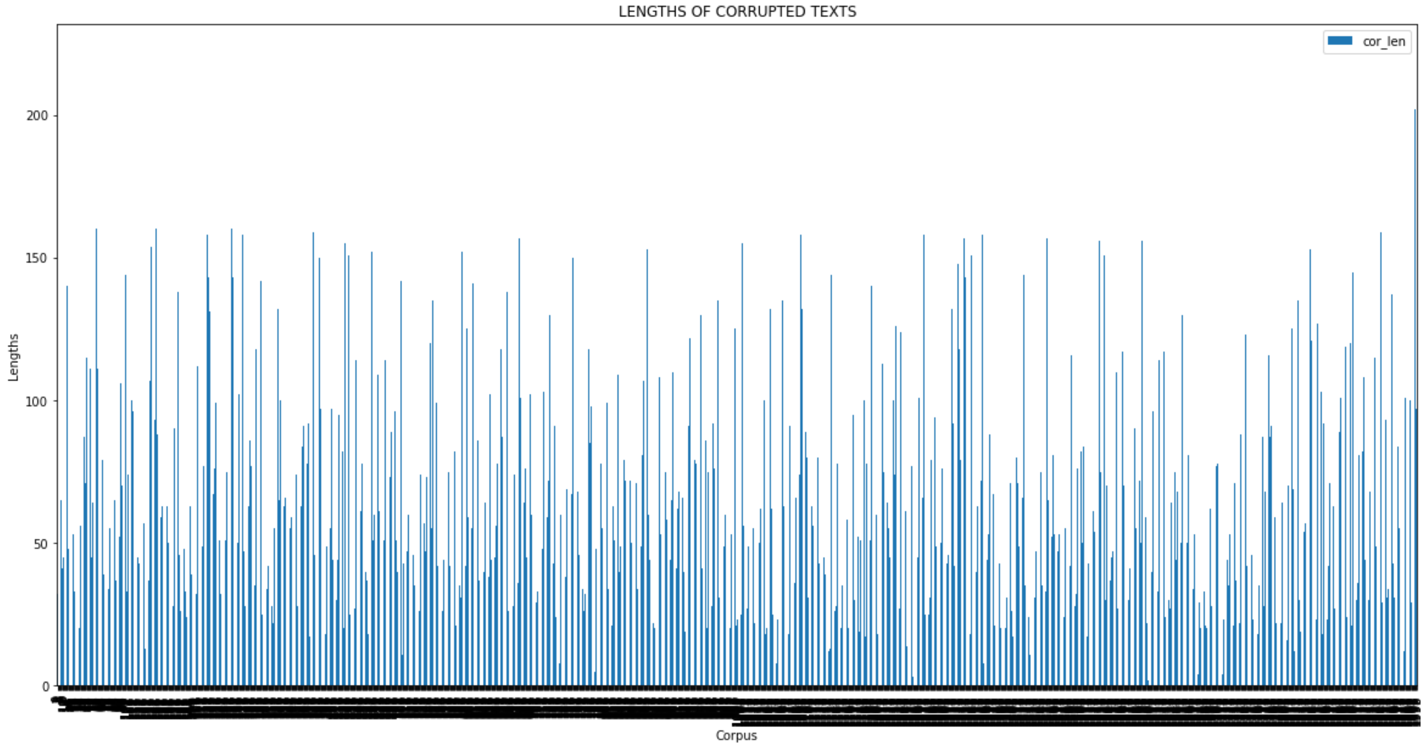
```
print('Max Length in Corrupted Texts:',final['cor_len'].max())
print('-'*100)

fig, ax = plt.subplots()
final['cor_len'].plot(ax=ax, kind='bar',figsize=(20, 10))

plt.xlabel('Corpus')
plt.ylabel('Lengths')
plt.title("LENGTHS OF CORRUPTED TEXTS")
```

```
ax.legend()
```

```
Max Length in Corrupted Texts: 221
----------------------------------------------------------------------------------------
-------------
```

Out[ ]:

```
<matplotlib.legend.Legend at 0x7f3950e27750>
```



PLOT DESCRIPTION AND OBSERVATIONS:

1.Here the bar plot showing lengths of all texts.

2.We have seen that we have mixed lengths of character texts, and highest length is= 221, and median length lies between 150-160.

3.Using this, we can decide the constant length of corrupted characters vector.

LENGTH OF ENGLISH TEXTS AT CHARACTER LEVEL

In [ ]:

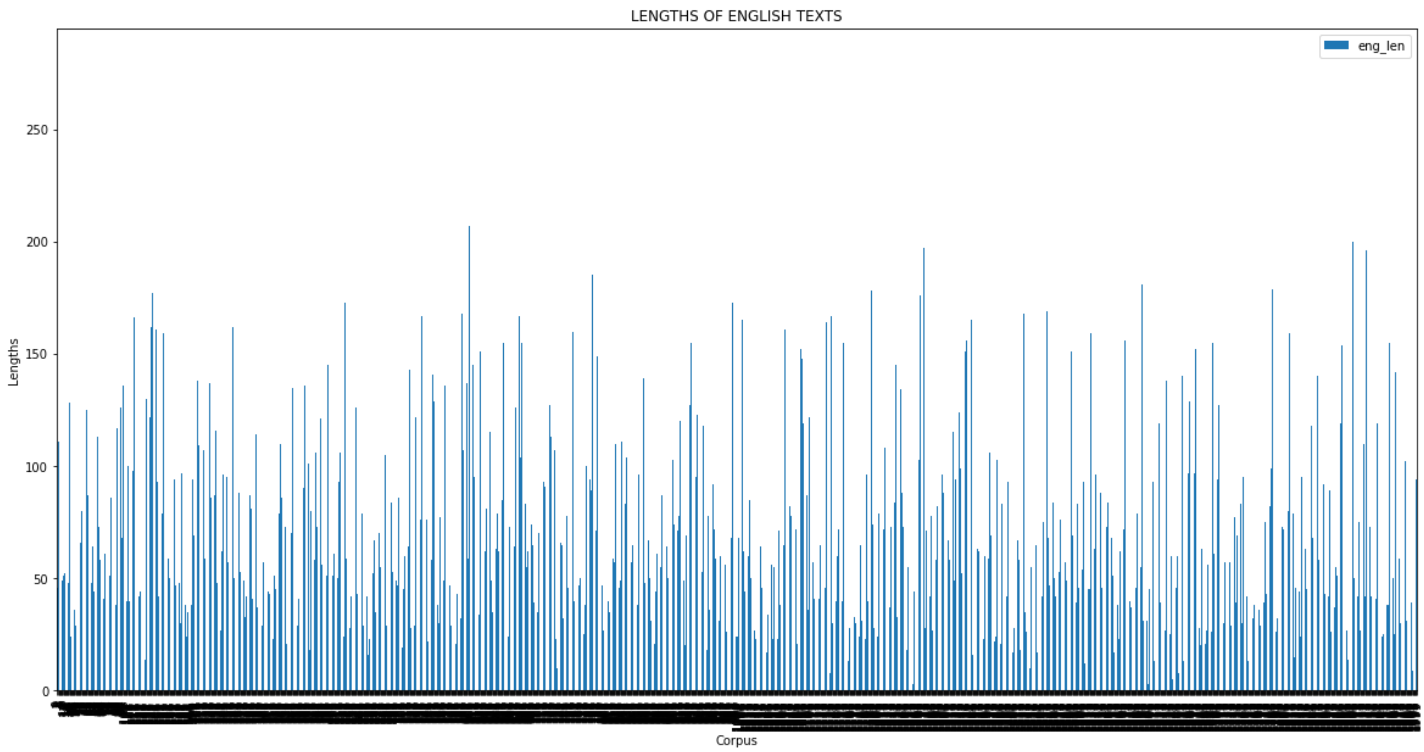
```
print('Max Length in English Texts:',final['eng_len'].max())
print('-'*100)

fig, ax = plt.subplots()
final['eng_len'].plot(ax=ax, kind='bar',figsize=(20, 10))

plt.xlabel('Corpus')
plt.ylabel('Lengths')
plt.title("LENGTHS OF ENGLISH TEXTS")

ax.legend()
```

```
Max Length in English Texts: 281
----------------------------------------------------------------------------------------
-------------
```

Out[ ]:

```
<matplotlib.legend.Legend at 0x7f0a0f3b6250>
```

```
<matplotlib.legend.Legend at 0x7f0a013b6250>
```



PLOT DESCRIPTION AND OBSERVATIONS:

1.Here the bar plot showing lengths of all english texts.

2.We have seen that we have mixed lengths of character texts, and highest length is= 281, and median of text lies between 160-180.

3.Using this, we can decide the constant length of english characters vector.

LENGTH OF CORRUPTED TEXTS AT WORD LEVEL

In [ ]:

```
cor_len_char=[]
eng_len_char=[]

for i in final.values:
  cor_len_char.append(len(i[0].split(' ')))
  eng_len_char.append(len(i[1].split(' ')))

final['cor_len_char']=cor_len_char #storing the corrupted text word wise
final['eng_len_char']=eng_len_char #storing the english text word wise
```

In [ ]:

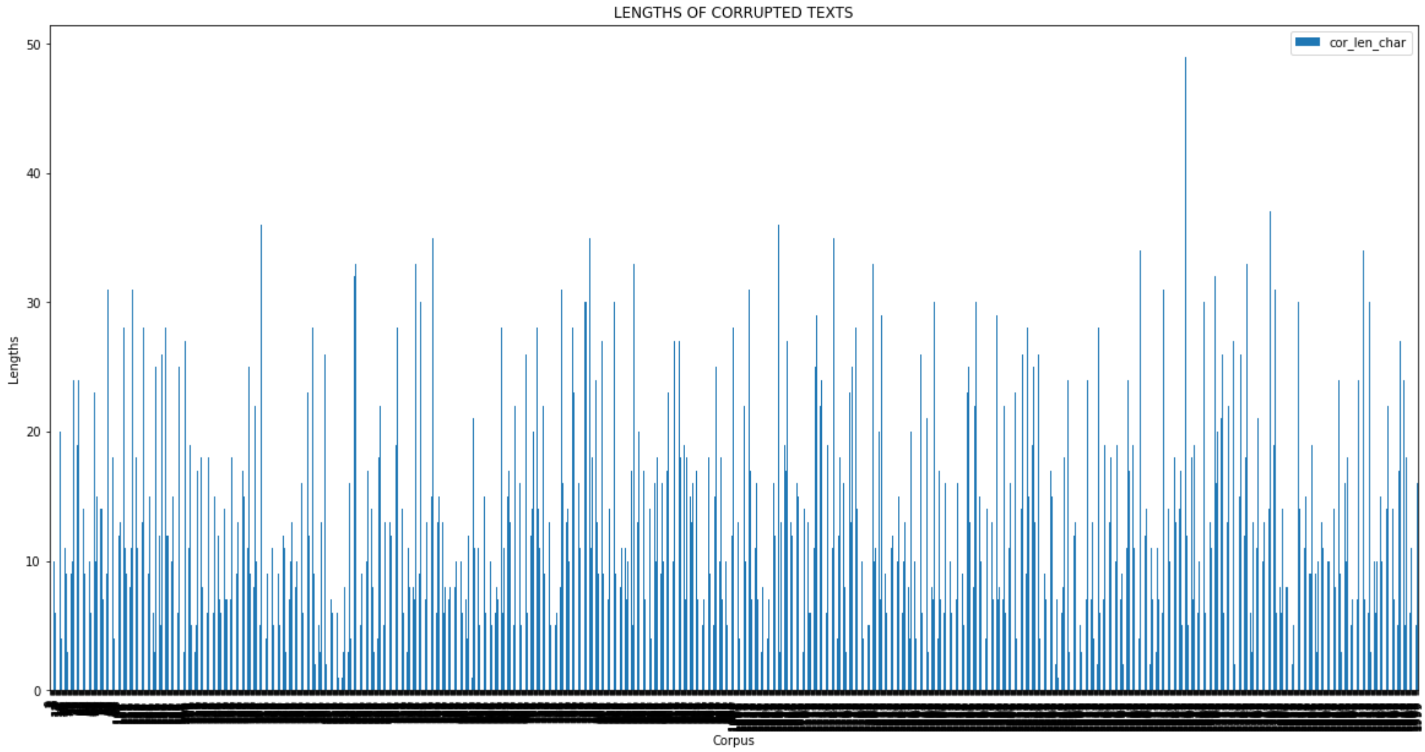
```
print('Max Length in Corrupted Texts:',final['cor_len_char'].max())
print('-'*100)

fig, ax = plt.subplots()
final['cor_len_char'].plot(ax=ax, kind='bar',figsize=(20, 10))

plt.xlabel('Corpus')
plt.ylabel('Lengths')
plt.title("LENGTHS OF CORRUPTED TEXTS")

ax.legend()
```

```
Max Length in Corrupted Texts: 49
----------------------------------------------------------------------------------------
-------------
```

```
-------------
```

Out[ ]:

```
<matplotlib.legend.Legend at 0x7f0a0e6e8e50>
```



PLOT DESCRIPTION AND OBSERVATIONS:

1.Here the bar plot showing lengths of all corrupted texts at word level.

2.We have seen that we have mixed lengths of word texts, and highest length is= 49, and median length lies between 30-36.

3.Using this, we can decide the constant length of corrupted word vector.

LENGTH OF ENGLISH TEXTS AT WORD LEVEL

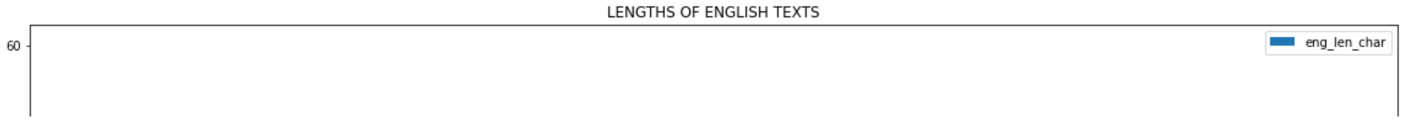
In [ ]:

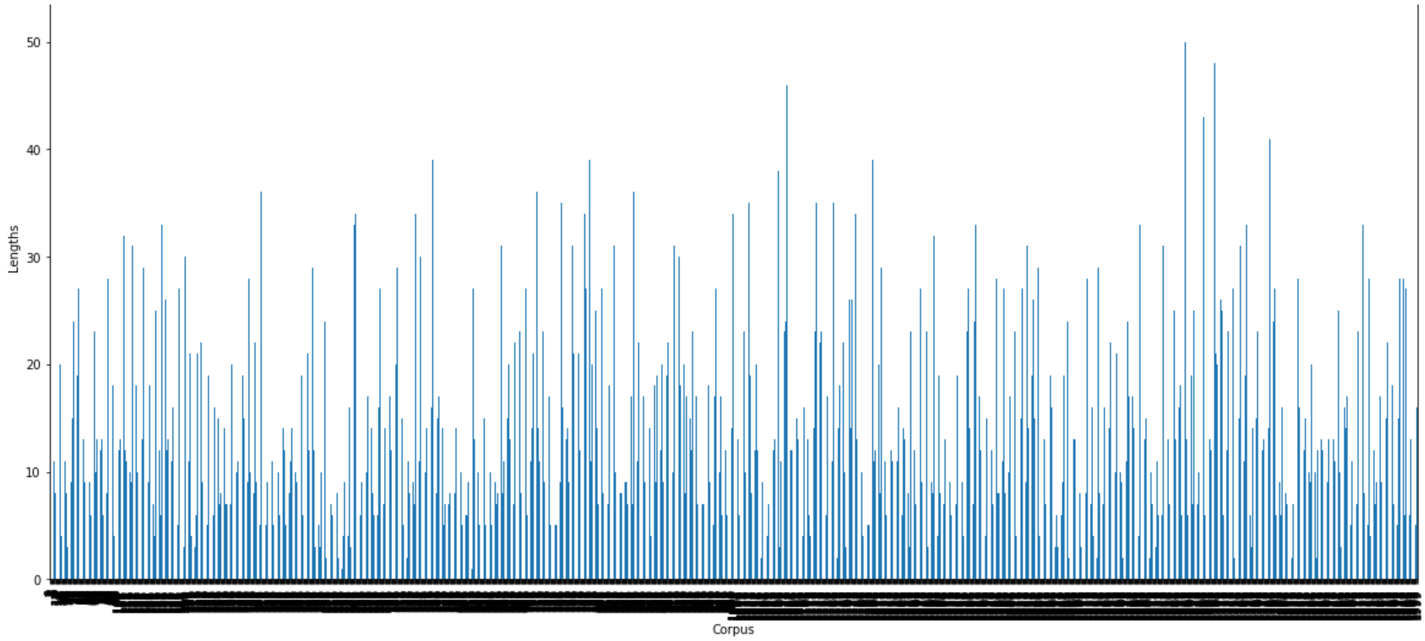
```python
print('Max Length in English Texts:',final['eng_len_char'].max())
print('-'*100)

fig, ax = plt.subplots()
final['eng_len_char'].plot(ax=ax, kind='bar',figsize=(20, 10))

plt.xlabel('Corpus')
plt.ylabel('Lengths')
plt.title("LENGTHS OF ENGLISH TEXTS")

ax.legend()
```

```
Max Length in English Texts: 59
---------------------------------------------------------------------------------------
-------------
```

Out[ ]:

```
<matplotlib.legend.Legend at 0x7f0a0daa5fd0>
```





PLOT DESCRIPTION AND OBSERVATIONS:

1.Here the bar plot showing lengths of all english texts at word level.

2.We have seen that we have mixed lengths of word texts, and highest length is= 59, and median length lies between 40-45.

3.Using this, we can decide the constant length of english word vector.

PERCENTILES AT WORD AND CHARCTER LEVEL LENGTHS OF TEXTS

In [ ]:

```
final.head()
```

Out[ ]:

	Corrupted	English	cor_len	eng_len	cor_len_char	eng_len_char
0	U wan me to "chop" seat 4 u nt?	Do you want me to reserve seat for you or not?	32	46	9	11
1	Yup. U reaching. We order some durian pastry a...	Yeap. You reaching? We ordered some Durian pas...	67	74	12	12
2	They become more ex oredi... Mine is like 25.....	They become more expensive already. Mine is li...	100	111	19	21
3	I'm thai. what do u do?	I'm Thai. What do you do?	23	25	6	6
4	Hi! How did your week go? Haven heard from you...	Hi! How did your week go? Haven't heard from y...	80	81	15	15

In [ ]:

```
for i in range(90,100,1):
   print('Percentile at:',i,'is:',np.percentile(final['cor_len'], i))

#As we have seen 99th percentile lies in 159, So we will chose our fixed character length based on this
```

```
Percentile at: 90 is: 125.0
Percentile at: 91 is: 129.0
Percentile at: 92 is: 134.0
Percentile at: 93 is: 138.0
Percentile at: 94 is: 142.0
Percentile at: 95 is: 147.0
Percentile at: 96 is: 151.0
Percentile at: 97 is: 154.0
Percentile at: 98 is: 157.0
Percentile at: 99 is: 159.0
```

In [ ]:

```
for i in range(90,100,1):
   print('Percentile at:',i,'is:',np.percentile(final['eng_len'], i))

#As we have seen 99th percentile lies in 190, So we will chose our fixed character length based on this
```

```
Percentile at: 90 is: 140.0
Percentile at: 91 is: 144.0
Percentile at: 92 is: 149.0
Percentile at: 93 is: 154.0
Percentile at: 94 is: 156.0
Percentile at: 95 is: 162.0
Percentile at: 96 is: 165.0
Percentile at: 97 is: 170.0
Percentile at: 98 is: 179.01999999999998
Percentile at: 99 is: 190.01
```

In [ ]:

```
for i in range(90,100,1):
   print('Percentile at:',i,'is:',np.percentile(final['cor_len_char'], i))

#As we have seen 99th percentile lies in 35, So we will chose our fixed word length based on this
```

```
Percentile at: 90 is: 25.0
Percentile at: 91 is: 26.0
Percentile at: 92 is: 27.0
Percentile at: 93 is: 28.0
Percentile at: 94 is: 28.0
Percentile at: 95 is: 29.0
Percentile at: 96 is: 30.0
Percentile at: 97 is: 31.0
Percentile at: 98 is: 33.0
Percentile at: 99 is: 35.0
```

In [ ]:

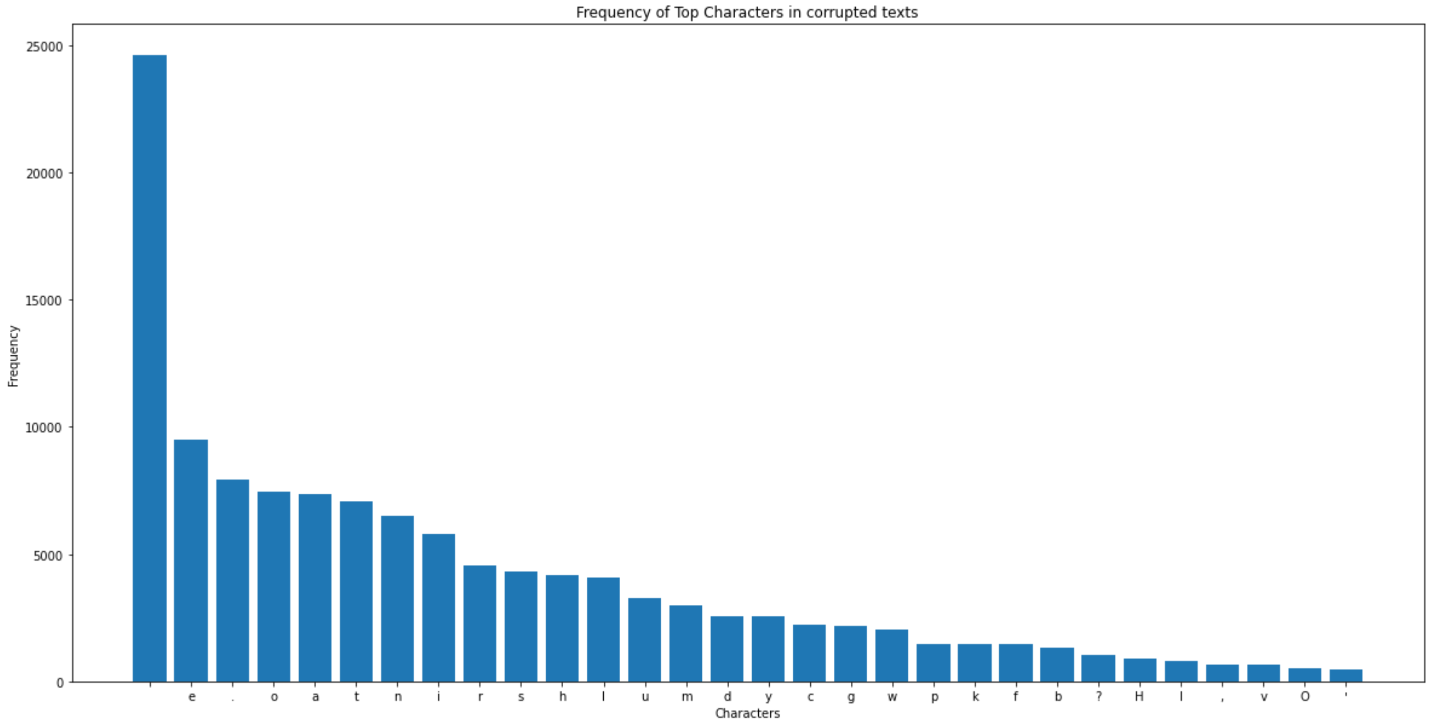
```
for i in range(90,100,1):
   print('Percentile at:',i,'is:',np.percentile(final['eng_len_char'], i))

#As we have seen 99th percentile lies in 38, So we will chose our fixed word length based on this
```

```
Percentile at: 90 is: 27.0
Percentile at: 91 is: 28.0
Percentile at: 92 is: 29.0
Percentile at: 93 is: 29.070000000000164
Percentile at: 94 is: 31.0
Percentile at: 95 is: 32.0
Percentile at: 96 is: 33.0
Percentile at: 97 is: 34.0
Percentile at: 98 is: 35.01999999999998
Percentile at: 99 is: 38.0
```

FREQUENCY OF CORRUPTED TEXT AT CHARACTER LEVEL

In [ ]:

```
import collections

corr_char=[]

for i in final['Corrupted'].values:
  for j in i:
    corr_char.append(str(j))

corr_freq=collections.Counter(corr_char) #creating a dictionary to store character/word a
t keys and their frequencies at values
corr_freq= dict(sorted(corr_freq.items(), key=lambda item: item[1])[::-1]) #sorting the d
ictionary based on values

ASCII = list(corr_freq.keys())[:30]
FREQEUNCY = list(corr_freq.values())[:30]

f, ax = plt.subplots(figsize=(20,10))
plt.bar(range(30),FREQEUNCY,tick_label=ASCII)

plt.xlabel('Characters')
plt.ylabel('Frequency')
plt.title("Frequency of Top Characters in corrupted texts")

plt.show()
```



PLOT DESCRIPTION AND OBSERVATIONS:

1.Here the Bar Plot shows the top frequency of characters in corrupted text.

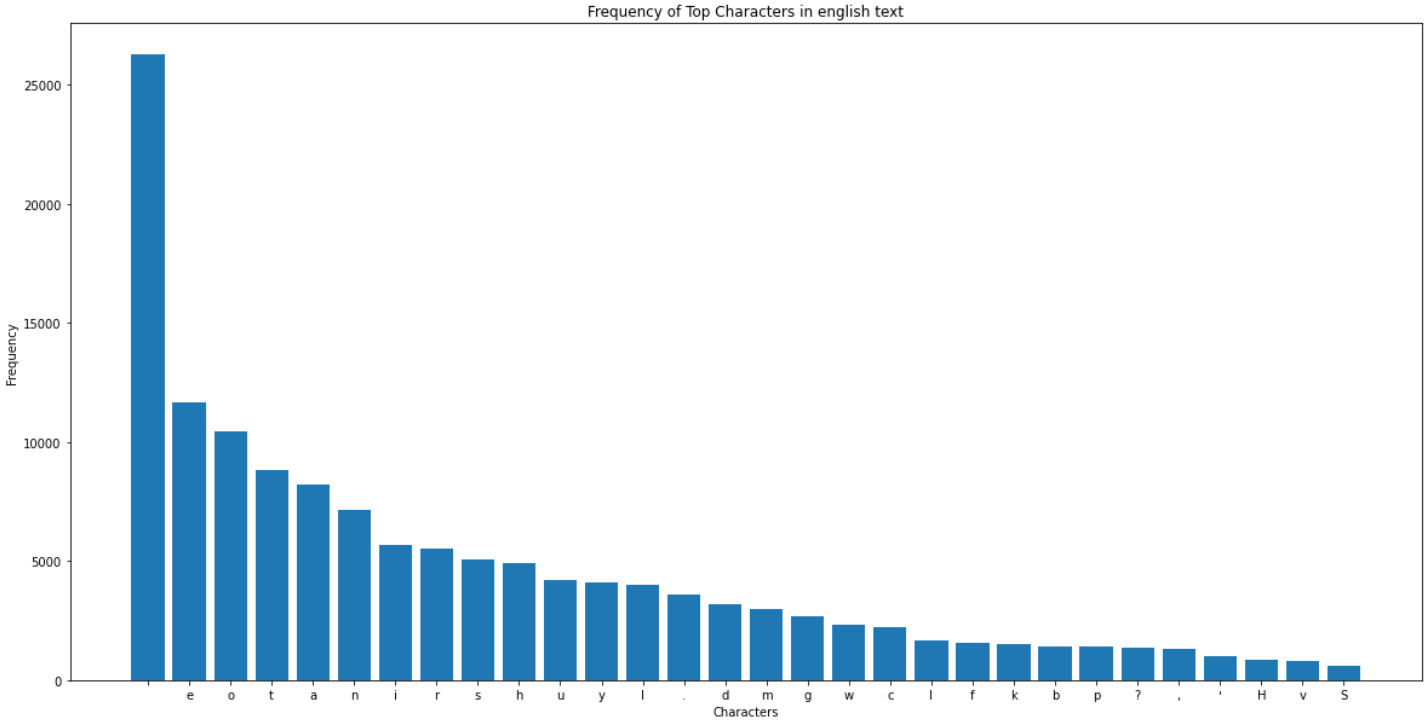
2.Blanks,e,.,a,t are the top characters that are used in the corpus.

3.The predictions of texts can be affected due to their high occurrences.

FREQUENCY OF ENGLISH TEXT AT CHARACTER LEVEL

In [ ]:

```
import collections

eng_char=[]

for i in final['English'].values:
  for j in i:
    eng_char.append(str(j))

corr_freq=collections.Counter(eng_char)
#creating a dictionary to store character/word at keys and their frequencies at values
corr_freq= dict(sorted(corr_freq.items(), key=lambda item: item[1])[::-1]) #sorting the d
ictionary based on values

ASCII = list(corr_freq.keys())[:30]
FREQEUNCY = list(corr_freq.values())[:30]

f, ax = plt.subplots(figsize=(20,10))
plt.bar(range(30),FREQEUNCY,tick_label=ASCII)

plt.xlabel('Characters')
plt.ylabel('Frequency')
plt.title("Frequency of Top Characters in english text")
plt.show()
```



PLOT DESCRIPTION AND OBSERVATIONS:

1.Here the Bar Plot shows the top frequency of characters in English text.

2.Blanks,e,o,t,a are the top characters that are used in the corpus.

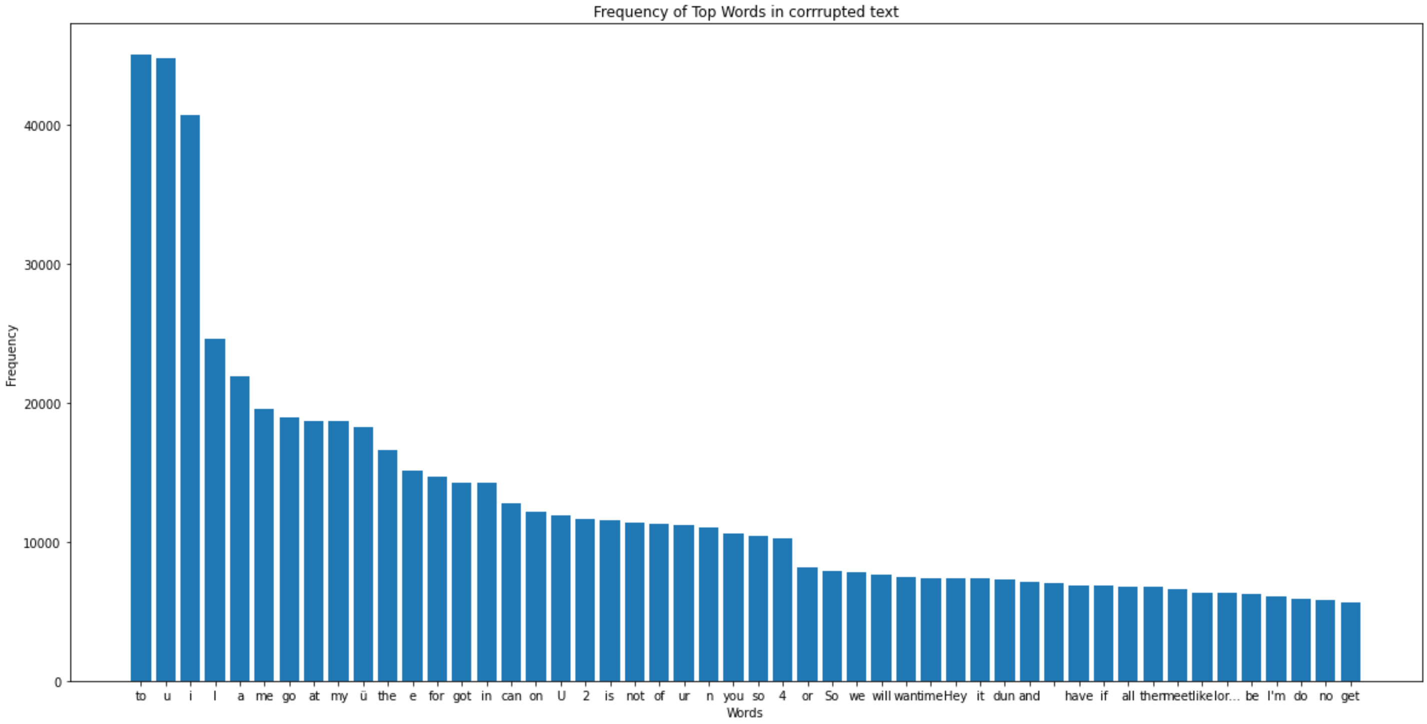
3.The predictions of texts can be affected due to their high occurrences.

FREQUENCY OF CORRUPTED TEXT AT WORD LEVEL

In [ ]:

```
import collections

corr_word=[]

for i in final['Corrupted'].values:
  for j in i:
    corr_word.extend(i.split(' '))

corr_freq=collections.Counter(corr_word)#creating a dictionary to store character/word at
keys and their frequencies at values
corr_freq= dict(sorted(corr_freq.items(), key=lambda item: item[1])[::-1])#sorting the di
```

```
ctionary based on values

ASCII = list(corr_freq.keys())[:50]
FREQEUNCY = list(corr_freq.values())[:50]

f, ax = plt.subplots(figsize=(20,10))
plt.bar(range(50),FREQEUNCY,tick_label=ASCII)

plt.xlabel('Words')
plt.ylabel('Frequency')
plt.title("Frequency of Top Words in corrrupted text")
plt.show()
```



PLOT DESCRIPTION AND OBSERVATIONS:

1.Here the Bar Plot shows the top frequency of words in corrupted text.

2.to,u,i,I,a are the top words that are used in the corpus.

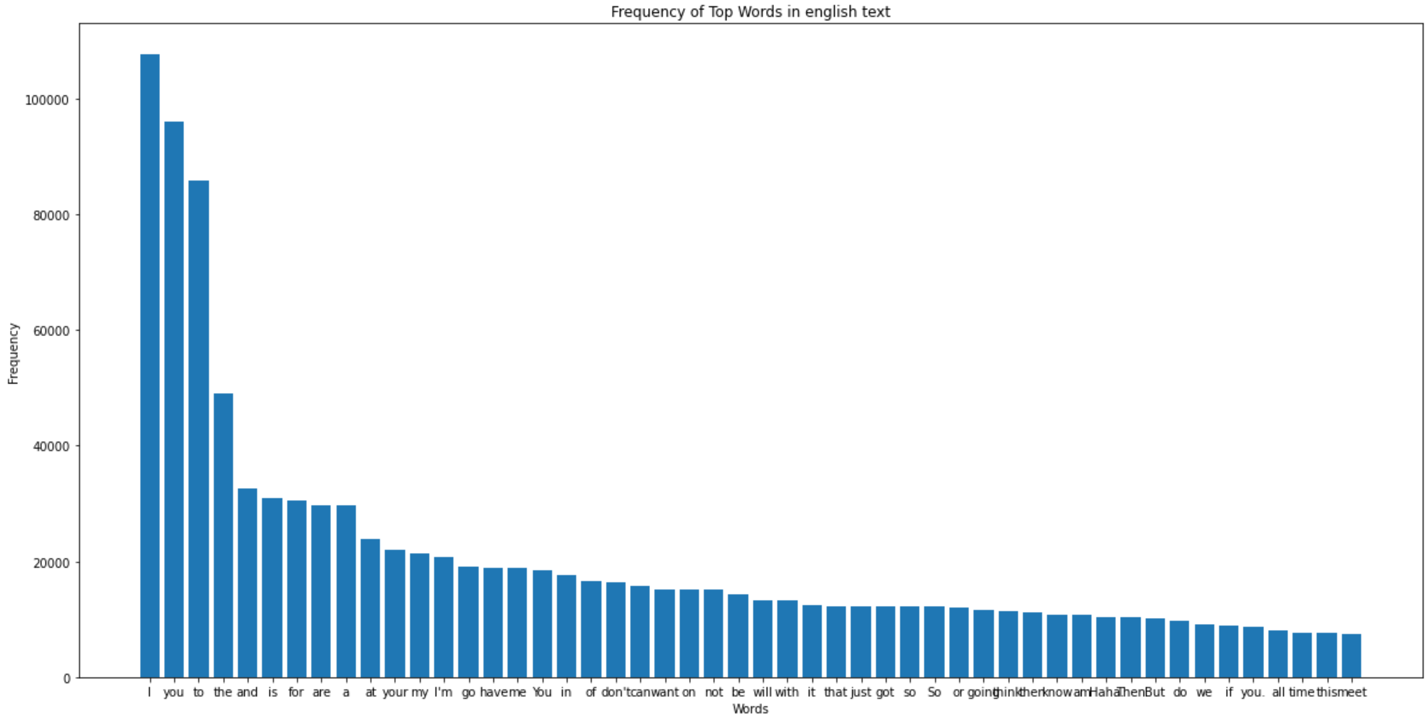
3.The predictions of texts can be affected due to their high occurrences.

FREQUENCY OF ENGLISH TEXT AT WORD LEVEL

In [ ]:

```
import collections
eng_word=[]

for i in final['English'].values:
  for j in i:
    eng_word.extend(i.split(' '))

corr_freq= collections.Counter(eng_word) #creating a dictionary to store character/word a
t keys and their frequencies at values
corr_freq= dict(sorted(corr_freq.items(), key=lambda item: item[1])[::-1]) #sorting the d
ictionary based on values

ASCII = list(corr_freq.keys())[:50]
FREQEUNCY = list(corr_freq.values())[:50]

f,ax = plt.subplots(figsize=(20,10))
plt.bar(range(50),FREQEUNCY,tick_label=ASCII)
```

```python
plt.xlabel('Words')
plt.ylabel('Frequency')
plt.title("Frequency of Top Words in english text")
plt.show()
```



PLOT DESCRIPTION AND OBSERVATIONS:

1.Here the Bar Plot shows the top frequency of words in english text.

2.I,you,to,the,and are the top words that are used in the corpus.

3.The predictions of texts can be affected due to their high occurrences.And In the corrected text, we can see some small single words like 'i' has been converted to 'I'.

3.Preprocessing Of Data

In [ ]:

```python
import pickle
final=pickle.load(open("/content/drive/MyDrive/CASE STUDY 2/final.pkl", "rb"))
final.head()
```

Out[ ]:

	Corrupted	English
0	U wan me to "chop" seat 4 u nt?	Do you want me to reserve seat for you or not?
1	Yup. U reaching. We order some durian pastry a...	Yeap. You reaching? We ordered some Durian pas...
2	They become more ex oredi... Mine is like 25.....	They become more expensive already. Mine is li...
3	I'm thai. what do u do?	I'm Thai. What do you do?
4	Hi! How did your week go? Haven heard from you...	Hi! How did your week go? Haven't heard from y...

A.Data Augmentation

As the Data is having only 2k points, We have performed the NLPAUG library based text augmentation of corrupted texts and their english.

In [ ]:

```
#installiing important augmentation libraries
#reference: https://pypi.org/project/nlpaug/

!pip3 install nlpaug
import nlpaug.augmenter.char as nac
import nlpaug.augmenter.word as naw
import nlpaug.augmenter.sentence as nas
import nlpaug.flow as nafc
import nltk
nltk.download('averaged_perceptron_tagger')
nltk.download('wordnet')
from nlpaug.util import Action
```

```
Collecting nlpaug
  Downloading nlpaug-1.1.7-py3-none-any.whl (405 kB)
     |████████████████████████████████| 405 kB 7.0 MB/s
Installing collected packages: nlpaug
Successfully installed nlpaug-1.1.7
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]     /root/nltk_data...
[nltk_data]   Unzipping taggers/averaged_perceptron_tagger.zip.
[nltk_data] Downloading package wordnet to /root/nltk_data...
[nltk_data]   Unzipping corpora/wordnet.zip.
```

In [ ]:

```
#first adding the orginal dataset in the starting
eng=[]
corrupt=[]

for i in final.values:
  eng.append(i[1])
  corrupt.append(i[0])
```

In [ ]:

```
for i in final['English'].values: #adding OCR corrupted Augmentation
  aug = naw.SynonymAug(aug_src='wordnet')
  eng.extend(2*[i])
  corrupt.extend(aug.augment(i,n=2))

for i in final['English'].values: #adding misspelled text augmentation
  aug = naw.SpellingAug()
  eng.extend(2*[i])
  corrupt.extend(aug.augment(i, n=2))
```

```
6000 6000
10000 10000
```

In [ ]:

```
import pandas as pd
corpus=pd.DataFrame()

#finally creating our dataset
corpus['Corrupted']=corrupt
corpus['English']=eng
```

In [ ]:

```
print('The shape of our final Dataset:',corpus.shape)
print('-'*100)
print(corpus.head(100))
```

```python
print('-'*100)
print(corpus.iloc[5600:5700,:])
print('-'*100)
print(corpus.tail(100))
```

```
The shape of our final Dataset: (10000, 2)
----------------------------------------------------------------------------------------
-------------
                                            Corrupted                                  
English
0                     U wan me to "chop" seat 4 u nt?     Do you want me to reserve seat f
or you or not?
1   Yup. U reaching. We order some durian pastry a...  Yeap. You reaching? We ordered som
e Durian pas...
2   They become more ex oredi... Mine is like 25.....  They become more expensive already
. Mine is li...
3                             I'm thai. what do u do?                         I'm Thai. 
What do you do?
4   Hi! How did your week go? Haven heard from you...  Hi! How did your week go? Haven't 
heard from y...
..                                                ...                                   
...
95  Yunny dear u wan to go for ur nus law bash ton...  Dear, you want to go for your NUS 
law bash ton...
96                      I go bathe first. U can ard 2.           I go bathing first. You 
can around two.
97  If lydat i cant do anyting.unless u already de...  If like that I can't do anyting. U
nless you al...
98  Mine was not. It said that the gal who sent it...  Mine was not. It said that the gir
l who sent i...
99                      got meh.. mine is local one...                           Got? Mi
ne is local one.

[100 rows x 2 columns]
----------------------------------------------------------------------------------------
-------------
                                              Corrupted                                 
English
5600                  Make that iii! For Deity ' s sake!                         Make that 
3! For God's sake!
5601                 Make that 3! For Divinity ' s sake!                         Make that 
3! For God's sake!
5602  I lie with. Some award display right? Haha. Me...  I know. Some award show right? H
aha. But I lik...
5603  1 roll in the hay. Some honour show right? Hah...  I know. Some award show right? H
aha. But I lik...
5604  I get into ' t understand her also. Then I enu...  I don't understand her also. The
n I said don't...
...                                                 ...                                 
...
5695  Sorry. Atomic number 53 didn ' t fuck you call...  Sorry. I didn't know you called. 
We are not ha...
5696               So how are you spending your weekend?             So how are you spend
ing your weekend?
5697                  So how are you spend your weekend?             So how are you spend
ing your weekend?
5698  One merely came back from Australia yesterday....  I just came back from Australia 
yesterday. Can...
5699  1 just come rearward from Australia yesterday....  I just came back from Australia 
yesterday. Can...

[100 rows x 2 columns]
----------------------------------------------------------------------------------------
-------------
                                              Corrupted                                 
English
9900                           Ah? Why is'll she n't going?                          Ah? Why 
is she not going?
9901                            Ah? Why i'ts she nit going?                          Ah? Why 
is she not going?
9902                                 Nopez. Nothing ad all.                            Nopez
. Nothing  at all.
```

```
9903                              Nopez. Nothing at always.                             Nopez
. Nothing  at all.
9904               U can come ENDE pick MY up anytime now.               You can come and pick m
e up anytime now.
...                                                   ...                                                 
...
9995                             We rare near Coca already.                             We are 
near Coca already.
9996   Hall eleven. Got lectures. end forges about co...   Hall eleven. Got lectures. And f
orget about co...
9997   Hall eleven. Got lectures. And forget about co...   Hall eleven. Got lectures. And f
orget about co...
9998   I bring for yoo. i''m can net promise you 100%...   I bring for you. I can not promi
se you 100% to...
9999   Im bring vor you. lI can not promese you 100% ...   I bring for you. I can not promi
se you 100% to...

[100 rows x 2 columns]
```

B.Getting Fixed Lengths

In [ ]:

```
#pickle.dump(corpus, open("/content/drive/MyDrive/CASE STUDY 2/latest.pkl","wb"))
```

In [ ]:

```
import pickle
corpus=pickle.load(open("/content/drive/MyDrive/CASE STUDY 2/latest.pkl", "rb"))
print(corpus.shape)
print('-'*100)
corpus.head()
```

```
(10000, 2)
----------------------------------------------------------------------------------------
-------------
```

Out[ ]:

	Corrupted	English
0	U wan me to "chop" seat 4 u nt?	Do you want me to reserve seat for you or not?
1	Yup. U reaching. We order some durian pastry a...	Yeap. You reaching? We ordered some Durian pas...
2	They become more ex oredi... Mine is like 25.....	They become more expensive already. Mine is li...
3	I'm thai. what do u do?	I'm Thai. What do you do?
4	Hi! How did your week go? Haven heard from you...	Hi! How did your week go? Haven't heard from y...

A. WORD LEVEL REDUCTION

In [ ]:

```
word_corr=[]
word_eng=[]

#As we have seen for corrupted word data and english word data
#35 and 38 lengths cover 99% of the data respectively, so we will make fixed lengths

for i in corpus.values:
  if len(str(i[0]).split(' '))<=35 and len(str(i[1]).split(' '))<=38:
    word_corr.append(i[0])
    word_eng.append(i[1])

word_final=pd.DataFrame()
word_final['corrupt_word']=word_corr
word_final['english_word']=word_eng

print('Shape of Word level dataframe: ',word_final.shape)
print('-'*100)
word_final.head()
```

```
Shape of Word level dataframe:  (9582, 2)
---------------------------------------------------------------------------------------
-------------
```

Out[ ]:

	corrupt_word	english_word
0	U wan me to "chop" seat 4 u nt?	Do you want me to reserve seat for you or not?
1	Yup. U reaching. We order some durian pastry a...	Yeap. You reaching? We ordered some Durian pas...
2	They become more ex oredi... Mine is like 25.....	They become more expensive already. Mine is li...
3	I'm thai. what do u do?	I'm Thai. What do you do?
4	Hi! How did your week go? Haven heard from you...	Hi! How did your week go? Haven't heard from y...

B. CHARACTER LEVEL REDUCTION

In [ ]:

```
char_corr=[]
char_eng=[]

#As we have seen for corrupted character data and english charcter data
#159 and 190 lengths cover 99% of the data respectively, so we will make fixed lengths

for i in corpus.values:
  if len(str(i[0]))<=159 and len(str(i[1]))<=190:
    char_corr.append(i[0])
    char_eng.append(i[1])

char_final=pd.DataFrame()
char_final['corrupt_char']=char_corr
char_final['english_char']=char_eng

print('Shape of Word level dataframe: ',char_final.shape)
print('-'*100)
char_final.head()
```

```
Shape of Word level dataframe:  (9265, 2)
---------------------------------------------------------------------------------------
-------------
```

Out[ ]:

	corrupt_char	english_char
0	U wan me to "chop" seat 4 u nt?	Do you want me to reserve seat for you or not?
1	Yup. U reaching. We order some durian pastry a...	Yeap. You reaching? We ordered some Durian pas...
2	They become more ex oredi... Mine is like 25.....	They become more expensive already. Mine is li...
3	I'm thai. what do u do?	I'm Thai. What do you do?
4	Hi! How did your week go? Haven heard from you...	Hi! How did your week go? Haven't heard from y...

In [ ]:

```
#pickle.dump(word_final, open("/content/drive/MyDrive/CASE STUDY 2/word_latest.pkl","wb")
)
#pickle.dump(char_final, open("/content/drive/MyDrive/CASE STUDY 2/char_latest.pkl","wb")
)
```

4.Feature Engineering and Getting Structured Data

In [ ]:

```
import pickle
word_final=pickle.load(open("/content/drive/MyDrive/CASE STUDY 2/word_latest.pkl", "rb"))
char_final=pickle.load(open("/content/drive/MyDrive/CASE STUDY 2/char_latest.pkl", "rb"))
```

In [ ]:

```
print('Word final dataframe:',word_final.shape)
print('Char final dataframe:',char_final.shape)
```

```
Word final dataframe: (9582, 2)
Char final dataframe: (9265, 2)
```

In [ ]:

```
word_final.head(2)
```

Out[ ]:

	corrupt_word	english_word
0	U wan me to "chop" seat 4 u nt?	Do you want me to reserve seat for you or not?
1	Yup. U reaching. We order some durian pastry a...	Yeap. You reaching? We ordered some Durian pas...

In [ ]:

```
#adding <start> tag at english input and <end> tag at english output
word_final['english_in'] = '<start> ' + word_final['english_word'].astype(str)
word_final['english_out'] = word_final['english_word'].astype(str) + ' <end>'

word_final = word_final.drop(['english_word'], axis=1)
word_final.head() #printing final word_dataframe
```

Out[ ]:

	corrupt_word	english_in	english_out
0	U wan me to "chop" seat 4 u nt?	<start> Do you want me to reserve seat for you...	Do you want me to reserve seat for you or not?...
1	Yup. U reaching. We order some durian pastry a...	<start> Yeap. You reaching? We ordered some Du...	Yeap. You reaching? We ordered some Durian pas...
2	They become more ex oredi... Mine is like 25.....	<start> They become more expensive already. Mi...	They become more expensive already. Mine is li...
3	I'm thai. what do u do?	<start> I'm Thai. What do you do?	I'm Thai. What do you do? <end>
4	Hi! How did your week go? Haven heard from you...	<start> Hi! How did your week go? Haven't hear...	Hi! How did your week go? Haven't heard from y...

In [ ]:

```
import re
```

```
def decontractions(phrase):
    """decontracted takes text and convert contractions into natural form.
     ref: https://stackoverflow.com/questions/19790188/expanding-english-language-contrac
tions-in-python/47091490#47091490"""
    # specific
    phrase = re.sub(r"won\'t", "will not", phrase)
    phrase = re.sub(r"can\'t", "can not", phrase)
    phrase = re.sub(r"won\’t", "will not", phrase)
    phrase = re.sub(r"can\’t", "can not", phrase)

    # general
    phrase = re.sub(r"n\'t", " not", phrase)
    phrase = re.sub(r"\'re", " are", phrase)
    phrase = re.sub(r"\'s", " is", phrase)
    phrase = re.sub(r"\'d", " would", phrase)
    phrase = re.sub(r"\'ll", " will", phrase)
    phrase = re.sub(r"\'t", " not", phrase)
    phrase = re.sub(r"\'ve", " have", phrase)
    phrase = re.sub(r"\'m", " am", phrase)

    phrase = re.sub(r"n\’t", " not", phrase)
    phrase = re.sub(r"\’re", " are", phrase)
    phrase = re.sub(r"\’s", " is", phrase)
    phrase = re.sub(r"\’d", " would", phrase)
    phrase = re.sub(r"\’ll", " will", phrase)
    phrase = re.sub(r"\’t", " not", phrase)
    phrase = re.sub(r"\’ve", " have", phrase)
    phrase = re.sub(r"\’m", " am", phrase)

    return phrase

def preprocess_corr(text):

    # use this function to remove the contractions: https://gist.github.com/anandborad/d4
10a49a493b56dace4f814ab5325bbd
    # remove all the spacial characters: except space ' '
    text = decontractions(text)
    text = re.sub('[^A-Za-z0-9 ]+', '', text)
    return text


def preprocess_eng(text):

    # remove the words betweent brakets ()
    # remove these characters: {'$', ')', '?', '"', '’', '.',  '°', '!', ';', '/', "'",
'€', '%', ':', ',', '('}
    # replace these spl characters with space: '\u200b', '\xa0', '-', '/'
    # we have found these characters after observing the data points, feel free to explor
e more and see if you can do find more
    # you are free to do more proprocessing
    # note that the model will learn better with better preprocessed data

    text = decontractions(text)
    text = re.sub('[$)\?"’.°!;\'€%:,(/]', '', text)
    text = re.sub('\u200b', ' ', text)
    text = re.sub('\xa0', ' ', text)
    text = re.sub('-', ' ', text)
    return text
```

In [ ]:

```
word_final['corrupt_word'] = word_final['corrupt_word'].apply(preprocess_corr)
word_final['english_in'] = word_final['english_in'].apply(preprocess_eng)
word_final['english_out'] = word_final['english_out'].apply(preprocess_eng)

word_final.head()
```

Out[ ]:

	corrupt_word	english_in	english_out
		<start> Do you want me to reserve seat	Do you want me to reserve seat for you or

	corrupt_word	english_in	english_out
0	U wan me to chop seat 4 u nt	for you	not
1	Yup U reaching We order some durian pastry alr...	<start> Yeap You reaching We ordered some Duri...	Yeap You reaching We ordered some Durian pastr...
2	They become more ex oredi Mine is like 25 So h...	<start> They become more expensive already Min...	They become more expensive already Mine is lik...
3	I am thai what do u do	<start> I am Thai What do you do	I am Thai What do you do <end>
4	Hi How did your week go Haven heard from you f...	<start> Hi How did your week go Have not heard...	Hi How did your week go Have not heard from yo...

In [ ]:

```
#pickle.dump(word_final, open("/content/drive/MyDrive/CASE STUDY 2/preprocessed_latest.pkl","wb"))
```

In [ ]:

```

```