In previous notebook we have generated email dataset now we shall prepare the dataset as per our requirement

In [1]:

```
from google.colab import drive
drive.mount('gdrive',force_remount=True)
```

```
Go to this URL in a browser: https://accounts.google.com/o/oauth2/aut
h?client_id=947318989803-6bn6qk8qdgf4n4g3pfee6491hc0brc4i.apps.googleu
sercontent.com&redirect_uri=urn%3aietf%3awg%3aoauth%3a2.0%3aoob&respon
se_type=code&scope=email%20https%3a%2f%2fwww.googleapis.com%2fauth%2fd
ocs.test%20https%3a%2f%2fwww.googleapis.com%2fauth%2fdrive%20https%3a%
2f%2fwww.googleapis.com%2fauth%2fdrive.photos.readonly%20https%3a%2f%2
fwww.googleapis.com%2fauth%2fpeopleapi.readonly (https://accounts.goog
le.com/o/oauth2/auth?client_id=947318989803-6bn6qk8qdgf4n4g3pfee6491hc
0brc4i.apps.googleusercontent.com&redirect_uri=urn%3aietf%3awg%3aoaut
h%3a2.0%3aoob&response_type=code&scope=email%20https%3a%2f%2fwww.googl
eapis.com%2fauth%2fdocs.test%20https%3a%2f%2fwww.googleapis.com%2faut
h%2fdrive%20https%3a%2f%2fwww.googleapis.com%2fauth%2fdrive.photos.rea
donly%20https%3a%2f%2fwww.googleapis.com%2fauth%2fpeopleapi.readonly)

Enter your authorization code:
··········
Mounted at gdrive
```

In [0]:

```
import os
import cv2
import json
import re
import shutil
import numpy as np
import tarfile
import pickle
from bs4 import BeautifulSoup
import sys
import joblib
from functools import reduce
import operator
import multiprocessing
import pandas as pd
import matplotlib.pyplot as plt
import random
import matplotlib.pyplot as plt
from matplotlib import patches
from itertools import chain
import datetime
from tqdm import tqdm
from zipfile import ZipFile
from google.colab.patches import cv2_imshow
import collections
from collections import Counter
from sklearn.model_selection import train_test_split

from sklearn.utils import shuffle
%matplotlib inline
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd
from mpl_toolkits.mplot3d import Axes3D
import numpy as np
from pathlib import Path
from skimage.io import imread
from skimage.color import label2rgb
```

In [3]:

```
final_data_points = pd.read_csv('gdrive/My Drive/google/final_data_processed_my_ema
final_data_points.shape
```

Out[3]:

```
(164, 6)
```

In [4]:

```
final_data_points
```

Out[4]:

	File_name	To	Subject	Previous_email	
0	emails_/yernagulahemanth/inbox/299.txt	yernagulahemanth	git hub	NaN	ma this
1	emails_/yernagulahemanth/inbox/412.txt	yernagulahemanth	how are you comminig	NaN	yern cas
2	emails_/yernagulahemanth/inbox/412.txt	hemanthcasestudy4	how are you comminig	hello mrs yernagulahemanth casestudy are will ...	yern i wil
3	emails_/yernagulahemanth/inbox/381.txt	yernagulahemanth	regarding deep learning project	NaN	y wo
4	emails_/yernagulahemanth/inbox/400.txt	yernagulahemanth	webinar	NaN	h this
...	...	...	...	...	
159	emails_/yernagulahemanth/sent/117.txt	yernagulahemanth	regarding sql assignment	thank you for your response on sat sep pm appl...	he th
160	emails_/yernagulahemanth/sent/77.txt	team	about final project	NaN	c r prob
161	emails_/yernagulahemanth/sent/77.txt	yernagulahemanth	about final project	can you explain more about the problem stateme...	it rec
162	emails_/yernagulahemanth/sent/116.txt	team	regarding sql assignment	NaN	y c w
163	emails_/yernagulahemanth/sent/116.txt	yernagulahemanth	regarding sql assignment	you need to be comfortable with writing nested...	tha re

164 rows × 6 columns

In [0]:

```
def data_for_model(index_,dataframe=0):
      '''
      Creates a dataframe such a way that each content  will be divided into x and
      if  x is first word of content then y will be  second word of content  i.e if
      x is ith word  then y will be (i+1)th word.Like  this we are going to assign
      one word to maximum 5 words
      '''
      x = []
      y = []
      to = []
      subject = []
      content = []
      file_nm = []
      prev_email = []

      type_      = []
      text = dataframe.Content.iloc[index_]
      for i in range(len(text)):


              s = 0
              e = i
              for j in range(5):

                      ee = i+1
                      p = ' '.join(text.split()[s:e+1])
                      q = ' '.join(text.split()[ee:ee+j+1])

                      if len(q) == 0:
                        continue
                      else:

                        # print(p,'-->',q)

                        x.append(p)
                        y.append(q)
                        to.append(dataframe.To.iloc[index_])
                        subject.append(dataframe.Subject.iloc[index_])
                        prev_email.append(dataframe.Previous_email.iloc[index_])
                        file_nm.append(dataframe.File_name.iloc[index_])
                        content.append(dataframe.Content.iloc[index_])
                        type_.append(dataframe.Type.iloc[index_])

      data_to_model = pd.DataFrame(columns=['To', 'Subject', 'Previous_email', 'Typ

      data_to_model['x']=x
      data_to_model['y']=y
      data_to_model['To'] = to
      data_to_model['File_nm'] = file_nm
      data_to_model['Subject'] = subject
      data_to_model['Previous_email'] = prev_email
      data_to_model['Content'] = content
      data_to_model['Type'] = type_
      return data_to_model
```

In [0]:

```
def multi_processing(df,nm):
      '''
      Given dataframe is passed to data_to_model  function and saves
      to joblib  folder  with particular name.
      '''

      data_to_model = pd.DataFrame()

      shape_ = df.shape[0]
      current_dataframe = df
      for i in tqdm(range(shape_),position=0):
          data_to_model = data_to_model.append(data_for_model(i,df))
          data_to_model.drop_duplicates()

      joblib.dump(data_to_model,'joblib/sample_'+nm)
```

In [0]:

```
for i in tqdm(range(0,164,40),position=0):

      multiprocess = multiprocessing.Manager()
      p1   =  multiprocessing.Process(target=multi_processing,args=(final_data_poin
      p2   =  multiprocessing.Process(target=multi_processing,args=(final_data_poin
      p3   =  multiprocessing.Process(target=multi_processing,args=(final_data_poin
      p4   =  multiprocessing.Process(target=multi_processing,args=(final_data_poin

      p1.start()
      p2.start()
      p3.start()
      p4.start()

      p1.join()
      p2.join()
      p3.join()
      p4.join()
```

```
100%|██████████| 10/10 [00:02<00:00,  3.70it/s]
100%|██████████| 10/10 [00:02<00:00,  4.13it/s]
100%|██████████| 10/10 [00:02<00:00,  3.34it/s]
100%|██████████| 10/10 [00:02<00:00,  3.62it/s]
100%|██████████| 10/10 [00:01<00:00,  8.03it/s]
100%|██████████| 10/10 [00:01<00:00,  8.44it/s]
100%|██████████| 10/10 [00:01<00:00,  6.39it/s]
100%|██████████| 10/10 [00:02<00:00,  4.37it/s]
 80%|████████  | 8/10 [00:01<00:00,  7.75it/s]
100%|██████████| 10/10 [00:01<00:00,  9.12it/s]
100%|██████████| 10/10 [00:01<00:00,  7.28it/s]
100%|██████████| 10/10 [00:01<00:00,  4.61it/s]
100%|██████████| 10/10 [00:01<00:00, 10.19it/s]
100%|██████████| 10/10 [00:01<00:00,  9.36it/s]
100%|██████████| 10/10 [00:01<00:00,  7.64it/s]
100%|██████████| 10/10 [00:01<00:00,  7.08it/s]
0it [00:00, ?it/s]
0it [00:00, ?it/s]

100%|██████████| 4/4 [00:00<00:00, 24.84it/s]
100%|██████████| 5/5 [00:08<00:00,  1.52s/it]
```

In [0]:

```
all_samples = []
for i in tqdm(os.listdir('joblib'),position = 0):
  if os.path.isfile('joblib/'+i):
    all_samples.append(joblib.load('joblib/'+i))

df_after_sample = pd.DataFrame()
df_after_sample = df_after_sample.append([i for i in all_samples])



df_after_sample.reset_index(inplace=True)
df_after_sample.drop(['index'],inplace=True,axis= 1)
df_after_sample
```

```
100%|██████████| 20/20 [00:00<00:00, 162.31it/s]
```

Out[86]:

	To	Subject	Previous_email	Type	x	y
0	yernagulahemanth	about final project	NaN	c__d	hello	yernagulahemanth
1	yernagulahemanth	about final project	NaN	c__d	hello	yernagulahemanth can
2	yernagulahemanth	about final project	NaN	c__d	hello	yernagulahemanth can you
3	yernagulahemanth	about final project	NaN	c__d	hello	yernagulahemanth can you explain
4	yernagulahemanth	about final project	NaN	c__d	hello	yernagulahemanth can you explain more
...	...	...	...	...	...	...
38865	yernagulahemanth	doubt regarding hstack	NaN	c__d	hello yernagulahemanth please update your code...	call
38866	yernagulahemanth	doubt regarding hstack	NaN	c__d	hello yernagulahemanth please update your code...	call
38867	yernagulahemanth	doubt regarding hstack	NaN	c__d	hello yernagulahemanth please update your code...	call
38868	yernagulahemanth	doubt regarding hstack	NaN	c__d	hello yernagulahemanth please update your code...	call

	To	Subject	Previous_email	Type	x	y
38869	yernagulahemanth	doubt regarding hstack	NaN	c__d	hello yernagulahemanth please update your code...	call

38870 rows × 8 columns

Lets delete duplicates in dataframe

In [0]:

```
print('Shape of data befor dropping duplicates:',df_after_sample.shape)
df_after_sample = df_after_sample.drop_duplicates()
print('Shape of data after dropping duplicates:',df_after_sample.shape)
```

```
Shape of data befor dropping duplicates: (38870, 8)
Shape of data after dropping duplicates: (37230, 8)
```

In [0]:

```
df_after_sample.index = [i for i in range(df_after_sample.shape[0])]
df_after_sample.to_csv('gdrive/My Drive/google/my_df_after_sample.csv',index=False)
```

In [0]:

```
df_after_sample = pd.read_csv('gdrive/My Drive/google/my_df_after_sample.csv')
```

In [0]:

```
df_after_sample
```

Out[89]:

	To	Subject	Previous_email	Type	x	y
0	yernagulahemanth	about final project	NaN	c__d	hello	yernagulahemanth
1	yernagulahemanth	about final project	NaN	c__d	hello	yernagulahemanth can
2	yernagulahemanth	about final project	NaN	c__d	hello	yernagulahemanth can you
3	yernagulahemanth	about final project	NaN	c__d	hello	yernagulahemanth can you explain
4	yernagulahemanth	about final project	NaN	c__d	hello	yernagulahemanth can you explain more
...	...	...	...	...	...	...
37225	yernagulahemanth	doubt regarding hstack	NaN	c__d	hello yernagulahemanth please update your code...	over the
37226	yernagulahemanth	doubt regarding hstack	NaN	c__d	hello yernagulahemanth please update your code...	over the call
37227	yernagulahemanth	doubt regarding hstack	NaN	c__d	hello yernagulahemanth please update your code...	the
37228	yernagulahemanth	doubt regarding hstack	NaN	c__d	hello yernagulahemanth please update your code...	the call
37229	yernagulahemanth	doubt regarding hstack	NaN	c__d	hello yernagulahemanth please update your code...	call

37230 rows × 8 columns

In [0]:

```
data = pd.read_csv('gdrive/My Drive/google/my_df_after_sample.csv')
```

In [0]:

```
data.head()
```

Out[92]:

	To	Subject	Previous_email	Type	x	y	
0	yernagulahemanth	about final project	NaN	c__d	hello	yernagulahemanth	emails_/yernagulahe
1	yernagulahemanth	about final project	NaN	c__d	hello	yernagulahemanth can	emails_/yernagulahe
2	yernagulahemanth	about final project	NaN	c__d	hello	yernagulahemanth can you	emails_/yernagulahe
3	yernagulahemanth	about final project	NaN	c__d	hello	yernagulahemanth can you explain	emails_/yernagulahe
4	yernagulahemanth	about final project	NaN	c__d	hello	yernagulahemanth can you explain more	emails_/yernagulahe

Since our main goal is to predict next word(s) when to,sunject,previous email(if any) some part of content is given so lets featurize in following way

Sentance	Output
This	is
This	is introduction
This	is introduction to
This	is introduction to my
This	is introduction to my project
This is	introduction
This is	introduction to
This is	introduction to my
This is	introduction to my
This is	introduction to my project
This is introduction	to
This is introduction	to my
This is introduction	to my project

Note:

Each part is seperated with their tags like < to >< sub >< prv >< cont >

In [0]:

```
tspc = [] #  combination of To, Subject, Previous Email(if any), content of email

for i in range(data.shape[0]):
  tspc.append('<to> ' + str(data.To.iloc[i])+' <prv> '+ str(data.Previous_email.ilo
```

In [0]:

```
final_data = pd.DataFrame(zip(tspc,data.y.values),columns=['x','y'])
final_data.head()
```

Out[94]:

	x	y
0	<to> yernagulahemanth <prv> nan <sub> about f...	yernagulahemanth
1	<to> yernagulahemanth <prv> nan <sub> about f...	yernagulahemanth can
2	<to> yernagulahemanth <prv> nan <sub> about f...	yernagulahemanth can you
3	<to> yernagulahemanth <prv> nan <sub> about f...	yernagulahemanth can you explain
4	<to> yernagulahemanth <prv> nan <sub> about f...	yernagulahemanth can you explain more

In [0]:

```
final_data.to_csv('gdrive/My Drive/google/final_data_my_emails.csv',index=False)
final_data.shape
```

Out[95]:

```
(37230, 2)
```

Lets apply models in next document

In [0]:

```

```