Configuration Manual

MSc Research Project
Data Analytics

Ramandeep Singh
Student ID: X21106053

School of Computing
National College of Ireland

Supervisor: Abdul Razzaq

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: Ramandeep Singh………………………………………………………………………………………

Student ID: X21106053……………………………………………………………………………………….……

Programme: Data Analytics……………………………………… **Year:** …2022………………..

Module: ……MSc Academic Internship………………………………………………………….………

Supervisor: Abdul Razzaq…………………………………………………………………….………

Submission Due Date: …15/12/2022……………………………………………………………………………….………

Project Title: Text Summarization using Sequence to Sequence ……….

Word Count: ………………361……………… **Page Count**…………………9………………….……..

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: ………Ramandeep Singh……………………………………………………………

Date: ……15/12/2022…………………………………………………………………………

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Ramandeep Singh

X21106053

1 Introduction

The below steps show the specifications, tools and steps that are needed to configure the code. Sentiment analysis and topic modelling has been performed using machine learning and deep learning also word vectorization is done.

2 System Specification

Following are the system configuration:

- Operating System: Windows 11
- Processor: Intel Core i5 8th Gen
- Hard Drive: 500SSD
- RAM: 8GB

3 Software Tools

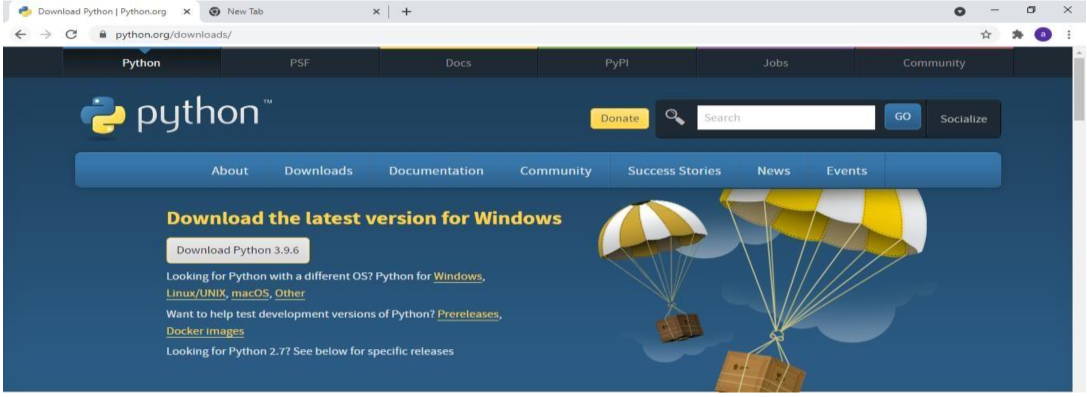
Some of the software tools used to implement this project are:

- Python
- Jupyter Notebook

3.1 Software Installation

This presents the processes taken in installing the tools used.

- Download and Installation of Python 3.9.6. The download link is https://www.python.org/downloads



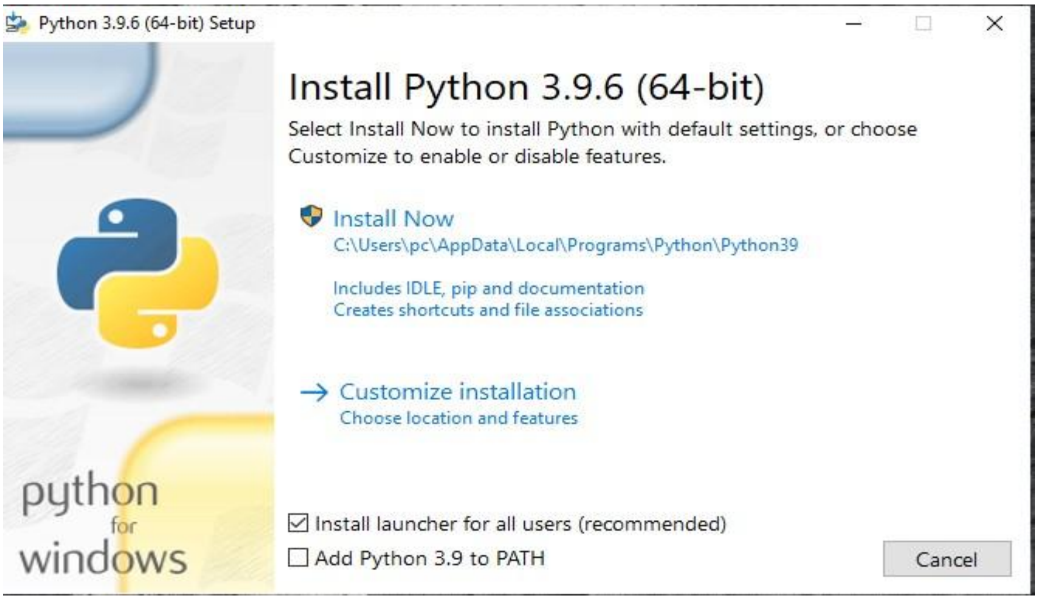
Download the latest version for Windows

Looking for Python with a different OS? Python for Windows, Linux/UNIX, macOS, Other

Want to help test development versions of Python? Prereleases, Docker images

Looking for Python 2.7? See below for specific releases

Fig 1: Python Download



Install Python 3.9.6 (64-bit)

Select Install Now to install Python with default settings, or choose Customize to enable or disable features.

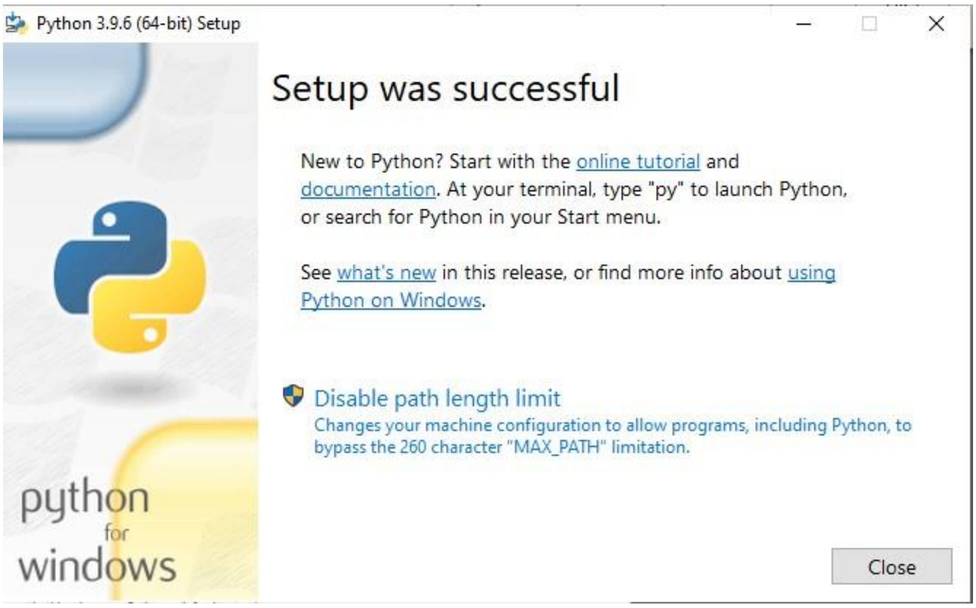
Install Now
C:\Users\pc\AppData\Local\Programs\Python\Python39

Includes IDLE, pip and documentation
Creates shortcuts and file associations

→ Customize installation
Choose location and features

☑ Install launcher for all users (recommended)
□ Add Python 3.9 to PATH

Fig 2: Python Installation



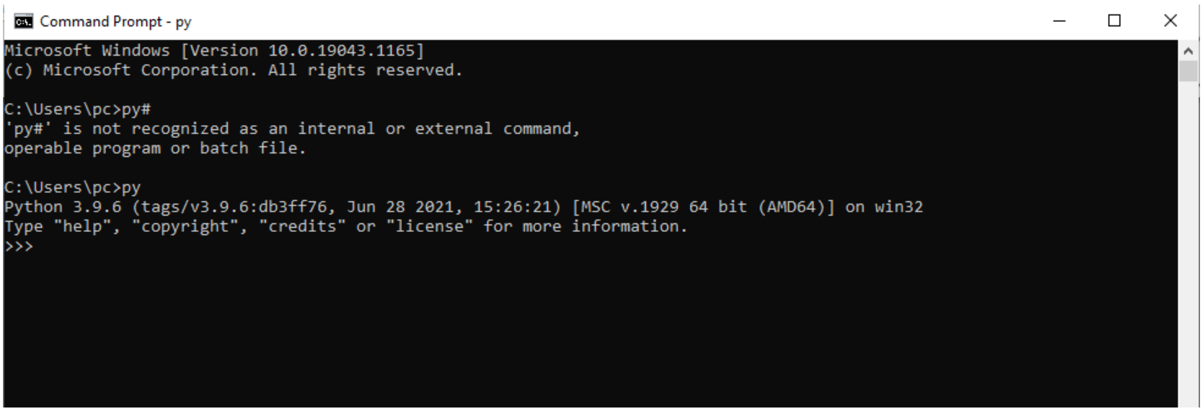
Setup was successful

New to Python? Start with the online tutorial and documentation. At your terminal, type "py" to launch Python, or search for Python in your Start menu.

See what's new in this release, or find more info about using Python on Windows.

Disable path length limit
Changes your machine configuration to allow programs, including Python, to bypass the 260 character "MAX_PATH" limitation.

Fig 3: Completion of Installation



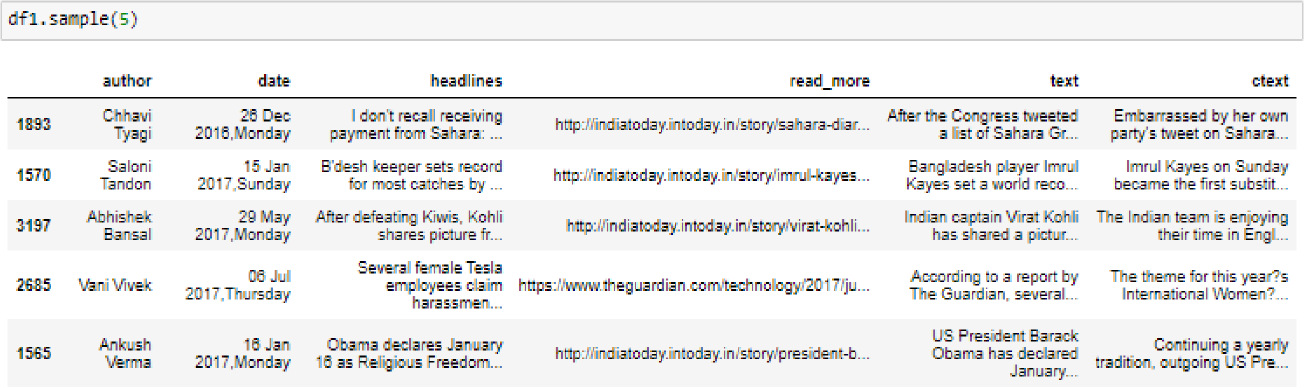
```
Microsoft Windows [Version 10.0.19043.1165]
(c) Microsoft Corporation. All rights reserved.

C:\Users\pc>py#
'py#' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\pc>py
Python 3.9.6 (tags/v3.9.6:db3ff76, Jun 28 2021, 15:26:21) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Fig 4: Confirmation of Python Installation

4 Implementation

The libraries from python used in implementing this project:

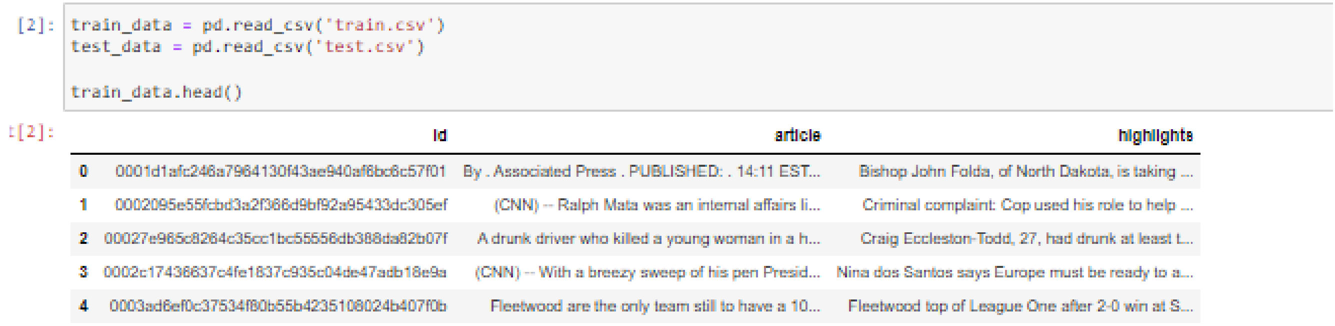
- Scikit-Learn
- Keras
- Pandas
- Pickle
- Numpy
- Genism
- Nltk
- Enchant
- Scacy
- Matplotlib
- Seaborn



```
df1.sample(5)
```

	author	date	headlines	read_more	text	ctext
1893	Chhavi Tyagi	26 Dec 2016,Monday	I don't recall receiving payment from Sahara: ...	http://indiatoday.intoday.in/story/sahara-diar...	After the Congress tweeted a list of Sahara Gr...	Embarrassed by her own party's tweet on Sahara...
1570	Saloni Tandon	15 Jan 2017,Sunday	B'desh keeper sets record for most catches by ...	http://indiatoday.intoday.in/story/imrul-kayes...	Bangladesh player Imrul Kayes set a world reco...	Imrul Kayes on Sunday became the first substit...
3197	Abhishek Bansal	29 May 2017,Monday	After defeating Kiwis, Kohli shares picture fr...	http://indiatoday.intoday.in/story/virat-kohli...	Indian captain Virat Kohli has shared a pictur...	The Indian team is enjoying their time in Engl...
2685	Vani Vivek	06 Jul 2017,Thursday	Several female Tesla employees claim harassmen...	https://www.theguardian.com/technology/2017/ju...	According to a report by The Guardian, several...	The theme for this year?s International Women?...
1565	Ankush Verma	16 Jan 2017,Monday	Obama declares January 16 as Religious Freedom...	http://indiatoday.intoday.in/story/president-b...	US President Barack Obama has declared January...	Continuing a yearly tradition, outgoing US Pre...

Fig 5: Checking the data on News Dataset

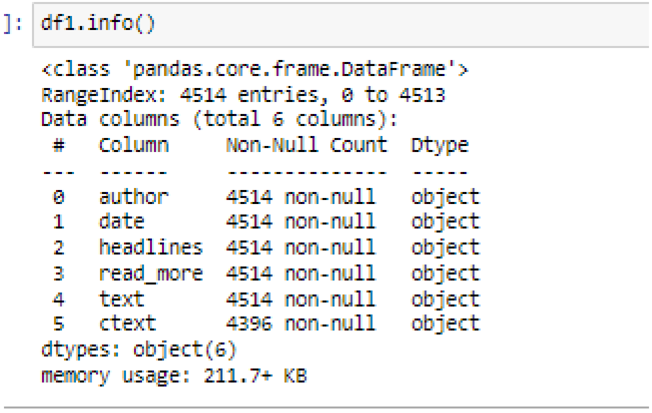


```
[2]: train_data = pd.read_csv('train.csv')
     test_data = pd.read_csv('test.csv')

     train_data.head()
```

t[2]:

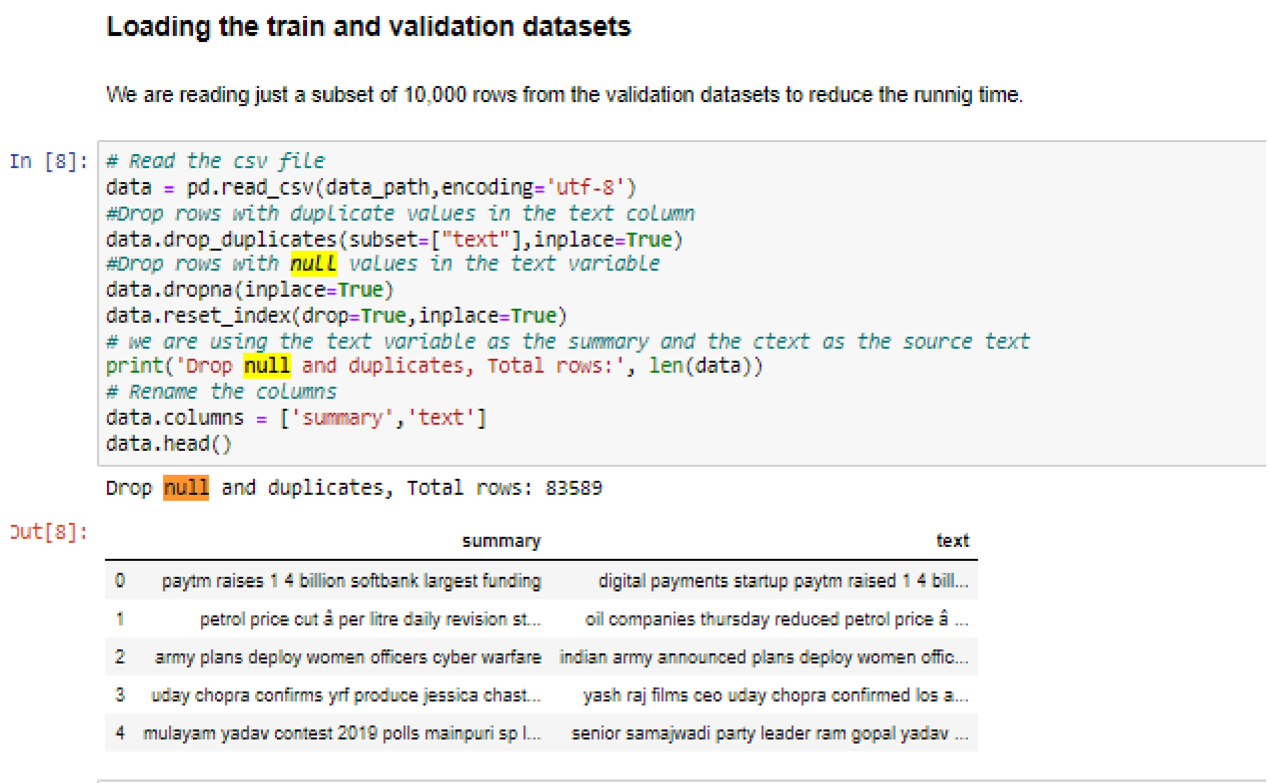
	Id	article	highlights
0	0001d1afc246a7964130f43ae940af6bc6c57f01	By . Associated Press . PUBLISHED: . 14:11 EST...	Bishop John Folda, of North Dakota, is taking ...
1	0002095e55fcbd3a2f366d9bf92a95433dc305ef	(CNN) -- Ralph Mata was an internal affairs li...	Criminal complaint: Cop used his role to help ...
2	00027e965c8264c35cc1bc55556db388da82b07f	A drunk driver who killed a young woman in a h...	Craig Eccleston-Todd, 27, had drunk at least t...
3	0002c17436637c4fe1837c935c04de47adb18e9a	(CNN) -- With a breezy sweep of his pen Presid...	Nina dos Santos says Europe must be ready to a...
4	0003ad6ef0c37534f80b55b4235108024b407f0b	Fleetwood are the only team still to have a 10...	Fleetwood top of League One after 2-0 win at S...

Fig 6: Checking the data on CNN/Daily News Dataset



```
]: df1.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4514 entries, 0 to 4513
Data columns (total 6 columns):
 #   Column      Non-Null Count  Dtype 
---  ------      --------------  ----- 
 0   author      4514 non-null   object
 1   date        4514 non-null   object
 2   headlines   4514 non-null   object
 3   read_more   4514 non-null   object
 4   text        4514 non-null   object
 5   ctext       4396 non-null   object
dtypes: object(6)
memory usage: 211.7+ KB
```

Fig 7: Data info on News Dataset



Loading the train and validation datasets

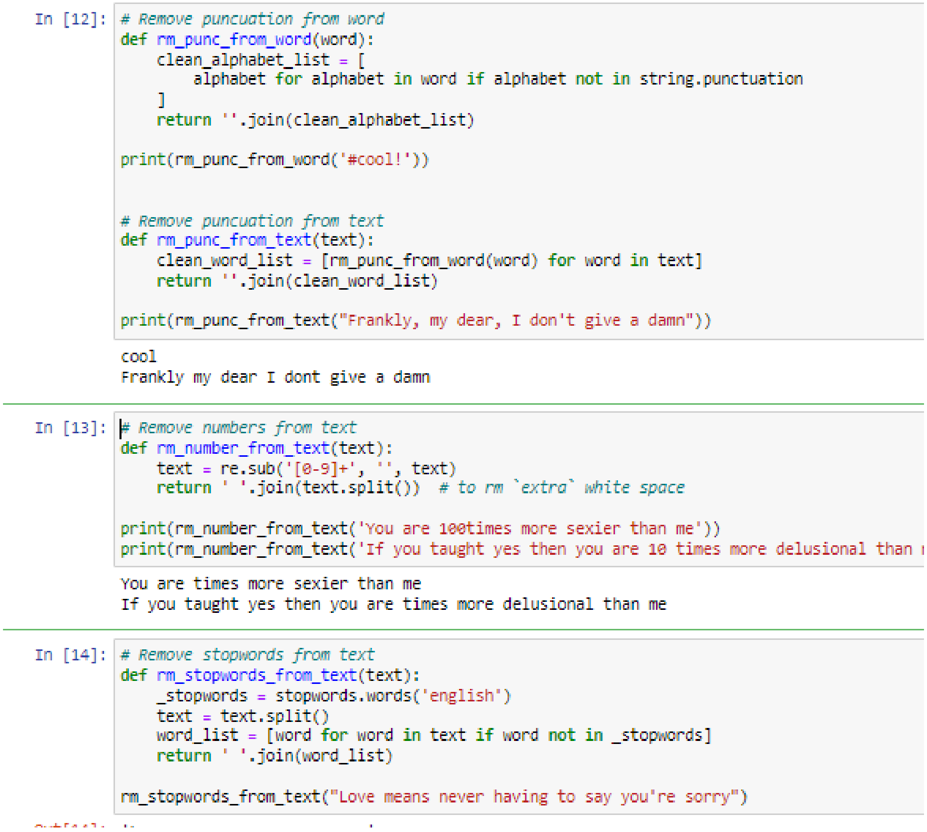
We are reading just a subset of 10,000 rows from the validation datasets to reduce the runnig time.

```
In [8]: # Read the csv file
        data = pd.read_csv(data_path,encoding='utf-8')
        #Drop rows with duplicate values in the text column
        data.drop_duplicates(subset=["text"],inplace=True)
        #Drop rows with null values in the text variable
        data.dropna(inplace=True)
        data.reset_index(drop=True,inplace=True)
        # we are using the text variable as the summary and the ctext as the source text
        print('Drop null and duplicates, Total rows:', len(data))
        # Rename the columns
        data.columns = ['summary','text']
        data.head()

        Drop null and duplicates, Total rows: 83589
```

Out[8]:

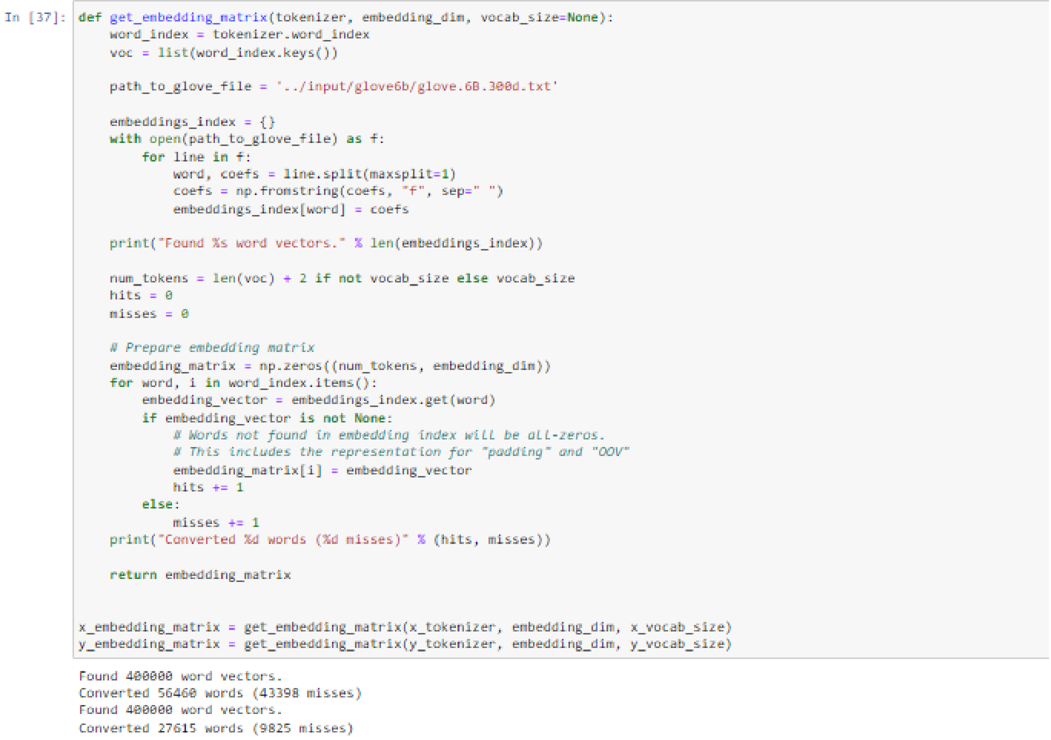
	summary	text
0	paytm raises 1 4 billion softbank largest funding	digital payments startup paytm raised 1 4 bill...
1	petrol price cut â per litre daily revision st...	oil companies thursday reduced petrol price â ...
2	army plans deploy women officers cyber warfare	indian army announced plans deploy women offic...
3	uday chopra confirms yrf produce jessica chast...	yash raj films ceo uday chopra confirmed los a...
4	mulayam yadav contest 2019 polls mainpuri sp l...	senior samajwadi party leader ram gopal yadav ...

Fig 8: Treating null values on News Dataset



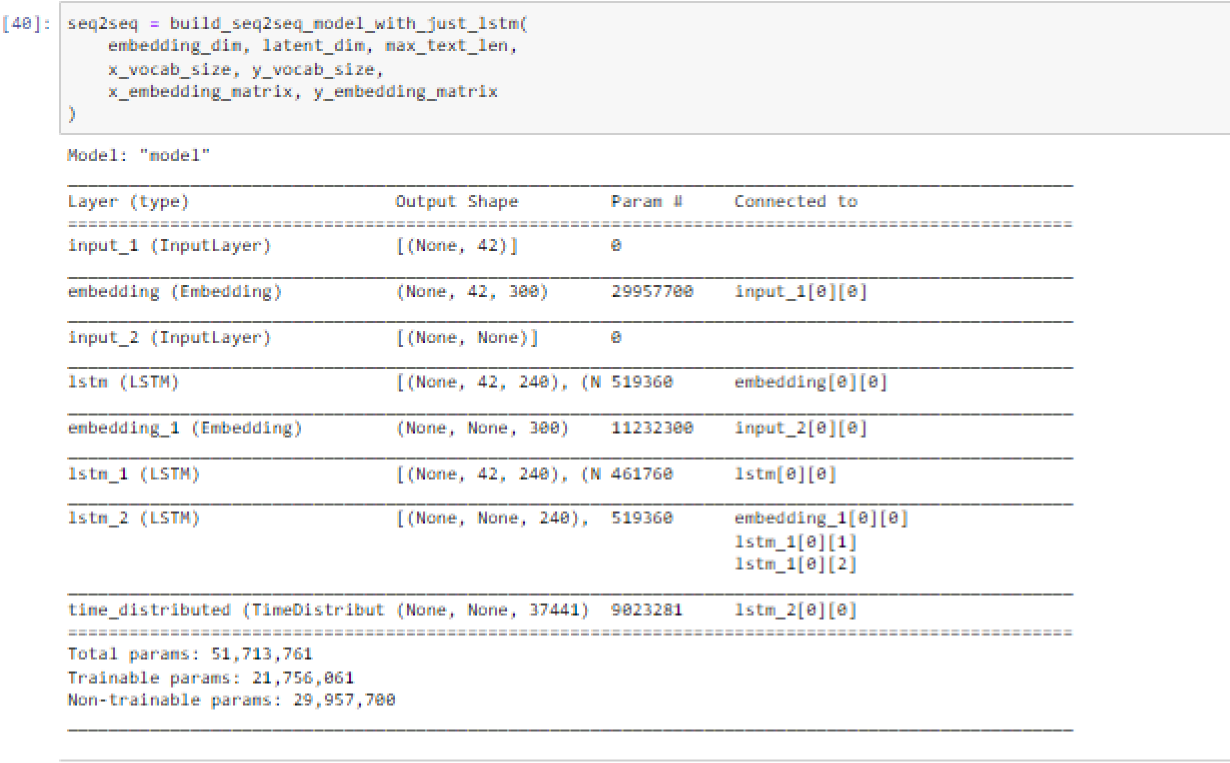
```
In [12]: # Remove puncuation from word
         def rm_punc_from_word(word):
             clean_alphabet_list = [
                 alphabet for alphabet in word if alphabet not in string.punctuation
             ]
             return ''.join(clean_alphabet_list)

         print(rm_punc_from_word('#cool!'))


         # Remove puncuation from text
         def rm_punc_from_text(text):
             clean_word_list = [rm_punc_from_word(word) for word in text]
             return ''.join(clean_word_list)

         print(rm_punc_from_text("Frankly, my dear, I don't give a damn"))

         cool
         Frankly my dear I dont give a damn

In [13]: # Remove numbers from text
         def rm_number_from_text(text):
             text = re.sub('[0-9]+', '', text)
             return ' '.join(text.split())  # to rm `extra` white space

         print(rm_number_from_text('You are 100times more sexier than me'))
         print(rm_number_from_text('If you taught yes then you are 10 times more delusional than 

         You are times more sexier than me
         If you taught yes then you are times more delusional than me

In [14]: # Remove stopwords from text
         def rm_stopwords_from_text(text):
             _stopwords = stopwords.words('english')
             text = text.split()
             word_list = [word for word in text if word not in _stopwords]
             return ' '.join(word_list)

         rm_stopwords_from_text("Love means never having to say you're sorry")
```

Fig 9: Removing punctuations, stop words, and special characters on News dataset



```
In [37]: def get_embedding_matrix(tokenizer, embedding_dim, vocab_size=None):
             word_index = tokenizer.word_index
             voc = list(word_index.keys())

             path_to_glove_file = '../input/glove6b/glove.6B.300d.txt'

             embeddings_index = {}
             with open(path_to_glove_file) as f:
                 for line in f:
                     word, coefs = line.split(maxsplit=1)
                     coefs = np.fromstring(coefs, "f", sep=" ")
                     embeddings_index[word] = coefs

             print("Found %s word vectors." % len(embeddings_index))

             num_tokens = len(voc) + 2 if not vocab_size else vocab_size
             hits = 0
             misses = 0

             # Prepare embedding matrix
             embedding_matrix = np.zeros((num_tokens, embedding_dim))
             for word, i in word_index.items():
                 embedding_vector = embeddings_index.get(word)
                 if embedding_vector is not None:
                     # Words not found in embedding index will be all-zeros.
                     # This includes the representation for "padding" and "OOV"
                     embedding_matrix[i] = embedding_vector
                     hits += 1
                 else:
                     misses += 1
             print("Converted %d words (%d misses)" % (hits, misses))

             return embedding_matrix


         x_embedding_matrix = get_embedding_matrix(x_tokenizer, embedding_dim, x_vocab_size)
         y_embedding_matrix = get_embedding_matrix(y_tokenizer, embedding_dim, y_vocab_size)

         Found 400000 word vectors.
         Converted 56460 words (43398 misses)
         Found 400000 word vectors.
         Converted 27615 words (9825 misses)
```

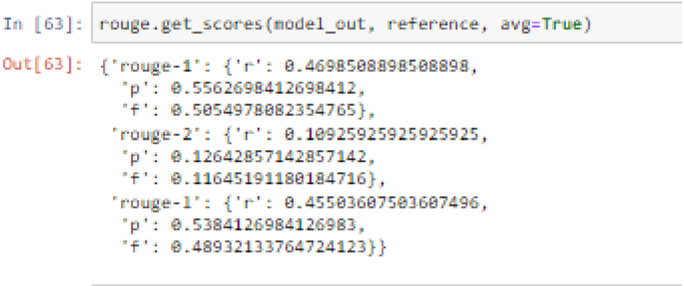
Fig 10: Word2vec Vectorization on News Dataset



```
[40]: seq2seq = build_seq2seq_model_with_just_lstm(
          embedding_dim, latent_dim, max_text_len,
          x_vocab_size, y_vocab_size,
          x_embedding_matrix, y_embedding_matrix
      )

      Model: "model"
      __________________________________________________________________________________________________
      Layer (type)                    Output Shape         Param #     Connected to
      ==================================================================================================
      input_1 (InputLayer)            [(None, 42)]         0
      __________________________________________________________________________________________________
      embedding (Embedding)           (None, 42, 300)      29957700    input_1[0][0]
      __________________________________________________________________________________________________
      input_2 (InputLayer)            [(None, None)]       0
      __________________________________________________________________________________________________
      lstm (LSTM)                     [(None, 42, 240), (N 519360      embedding[0][0]
      __________________________________________________________________________________________________
      embedding_1 (Embedding)         (None, None, 300)    11232300    input_2[0][0]
      __________________________________________________________________________________________________
      lstm_1 (LSTM)                   [(None, 42, 240), (N 461760      lstm[0][0]
      __________________________________________________________________________________________________
      lstm_2 (LSTM)                   [(None, None, 240),  519360      embedding_1[0][0]
                                                                       lstm_1[0][1]
                                                                       lstm_1[0][2]
      __________________________________________________________________________________________________
      time_distributed (TimeDistribut (None, None, 37441)  9023281     lstm_2[0][0]
      ==================================================================================================
      Total params: 51,713,761
      Trainable params: 21,756,061
      Non-trainable params: 29,957,700
      __________________________________________________________________________________________________
```

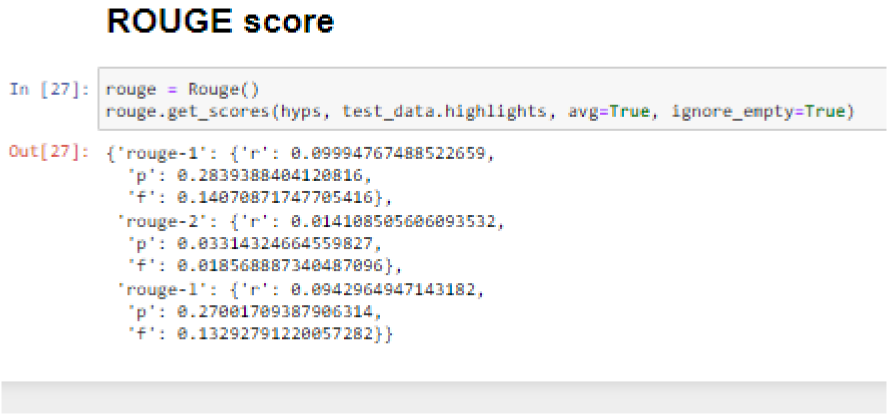
Fig 11: Building LSTM on News Dataset

5 EVALUATION:



```
In [63]: rouge.get_scores(model_out, reference, avg=True)

Out[63]: {'rouge-1': {'r': 0.4698508898508898,
            'p': 0.5562698412698412,
            'f': 0.5054978082354765},
           'rouge-2': {'r': 0.10925925925925925,
            'p': 0.12642857142857142,
            'f': 0.11645191180184716},
           'rouge-l': {'r': 0.45503607503607496,
            'p': 0.5384126984126983,
            'f': 0.48932133764724123}}
```

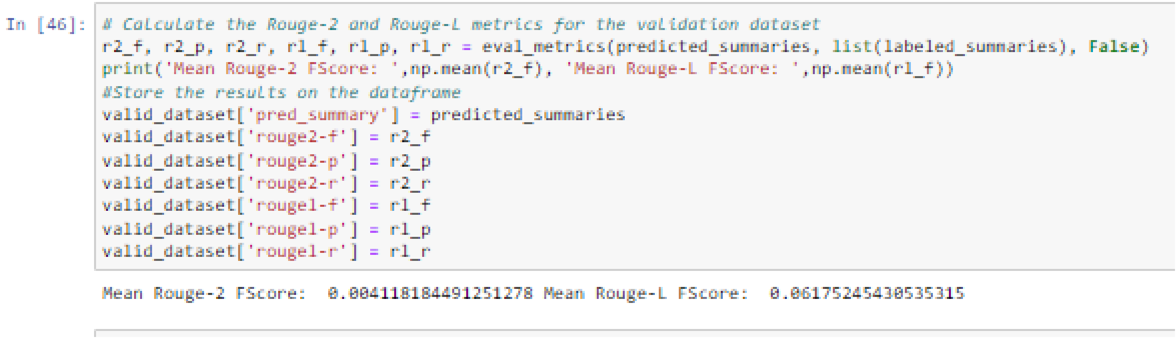
Fig 12: ROUGE Metrics of LSTM on News Dataset



ROUGE score

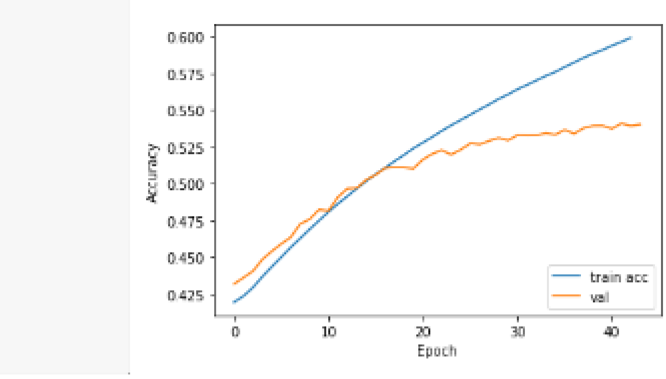
```
In [27]: rouge = Rouge()
         rouge.get_scores(hyps, test_data.highlights, avg=True, ignore_empty=True)

Out[27]: {'rouge-1': {'r': 0.09994767488522659,
            'p': 0.2839388404120816,
            'f': 0.1407087174770541 6},
           'rouge-2': {'r': 0.014108505606093532,
            'p': 0.03314324664559827,
            'f': 0.018568887340487096},
           'rouge-l': {'r': 0.0942964947143182,
            'p': 0.27001709387906314,
            'f': 0.13292791220057282}}
```

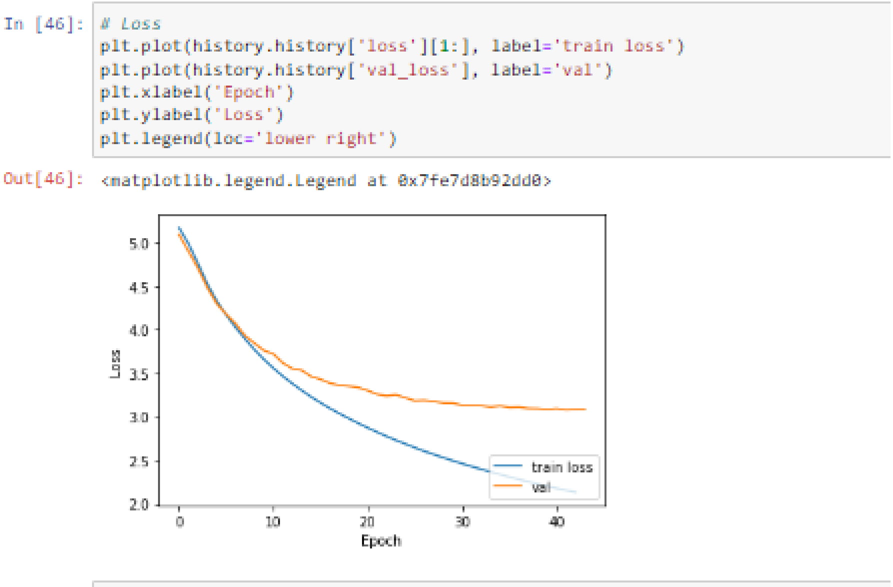
Fig 13: ROUGE Metrics of LSTM on CNN/Daily Mail Dataset



```
In [46]: # Calculate the Rouge-2 and Rouge-L metrics for the validation dataset
         r2_f, r2_p, r2_r, r1_f, r1_p, r1_r = eval_metrics(predicted_summaries, list(labeled_summaries), False)
         print('Mean Rouge-2 FScore: ',np.mean(r2_f), 'Mean Rouge-L FScore: ',np.mean(r1_f))
         #Store the results on the dataframe
         valid_dataset['pred_summary'] = predicted_summaries
         valid_dataset['rouge2-f'] = r2_f
         valid_dataset['rouge2-p'] = r2_p
         valid_dataset['rouge2-r'] = r2_r
         valid_dataset['rouge1-f'] = r1_f
         valid_dataset['rouge1-p'] = r1_p
         valid_dataset['rouge1-r'] = r1_r

         Mean Rouge-2 FScore:  0.004118184491251278 Mean Rouge-L FScore:  0.06175245430535315
```

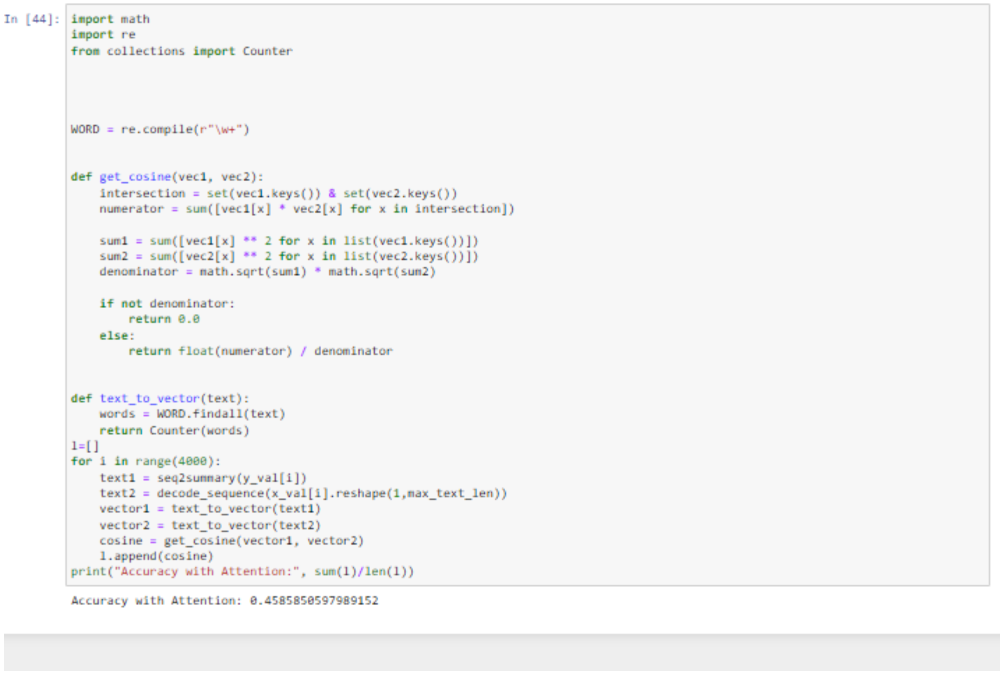
Fig 14: Mean ROUGE Metrics of attention mechanism





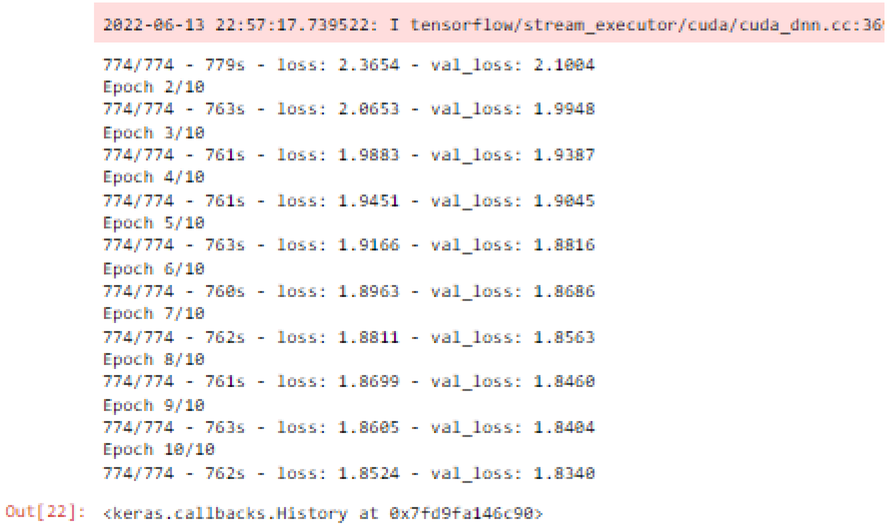
```
In [46]: # Loss
         plt.plot(history.history['loss'][1:], label='train loss')
         plt.plot(history.history['val_loss'], label='val')
         plt.xlabel('Epoch')
         plt.ylabel('Loss')
         plt.legend(loc='lower right')

Out[46]: <matplotlib.legend.Legend at 0x7fe7d8b92dd0>
```

Fig 15: LSTM Loss and Accuracy on News Dataset



```
In [44]: import math
         import re
         from collections import Counter



         WORD = re.compile(r"\w+")


         def get_cosine(vec1, vec2):
             intersection = set(vec1.keys()) & set(vec2.keys())
             numerator = sum([vec1[x] * vec2[x] for x in intersection])

             sum1 = sum([vec1[x] ** 2 for x in list(vec1.keys())])
             sum2 = sum([vec2[x] ** 2 for x in list(vec2.keys())])
             denominator = math.sqrt(sum1) * math.sqrt(sum2)

             if not denominator:
                 return 0.0
             else:
                 return float(numerator) / denominator


         def text_to_vector(text):
             words = WORD.findall(text)
             return Counter(words)
         l=[]
         for i in range(4000):
             text1 = seq2summary(y_val[i])
             text2 = decode_sequence(x_val[i].reshape(1,max_text_len))
             vector1 = text_to_vector(text1)
             vector2 = text_to_vector(text2)
             cosine = get_cosine(vector1, vector2)
             l.append(cosine)
         print("Accuracy with Attention:", sum(l)/len(l))

         Accuracy with Attention: 0.4585850597989152
```

Fig 16: LSTM with an attention mechanism accuracy



```
        2022-06-13 22:57:17.739522: I tensorflow/stream_executor/cuda/cuda_dnn.cc:36
        774/774 - 779s - loss: 2.3654 - val_loss: 2.1004
        Epoch 2/10
        774/774 - 763s - loss: 2.0653 - val_loss: 1.9948
        Epoch 3/10
        774/774 - 761s - loss: 1.9883 - val_loss: 1.9387
        Epoch 4/10
        774/774 - 761s - loss: 1.9451 - val_loss: 1.9045
        Epoch 5/10
        774/774 - 763s - loss: 1.9166 - val_loss: 1.8816
        Epoch 6/10
        774/774 - 760s - loss: 1.8963 - val_loss: 1.8686
        Epoch 7/10
        774/774 - 762s - loss: 1.8811 - val_loss: 1.8563
        Epoch 8/10
        774/774 - 761s - loss: 1.8699 - val_loss: 1.8460
        Epoch 9/10
        774/774 - 763s - loss: 1.8605 - val_loss: 1.8404
        Epoch 10/10
        774/774 - 762s - loss: 1.8524 - val_loss: 1.8340
Out[22]: <keras.callbacks.History at 0x7fd9fa146c90>
```

Fig 17: Epochs on CNN/Daily Mail News Dataset

6 Execution of the code

Following are the steps to run the code:-

1. Download the IDT zip by datasets and python files

2. unzip the files into a folder

3. Open Spyder or Jupyter through anaconda GUI

4. Open .py files in Spyder and .ipynb file in Jupyter

5. Change the path to path referencing dataset with in your folder

6. Run the code, either step by step or whole code at the same time.