Mandate 4

Final Report

Abstractive Text Summarization

Aakanksha Rani (MT2022001)

▼ Problem Statement :

Given a set of tweets pertaining to a trending topic, create an abstractive prose summary of the tweets. Do not just string the tweets together to form the summary. The summary will need to paraphrase and/or say more than what is directly said in the tweets. Propose a rubric to evaluate the accuracy of your summarization.

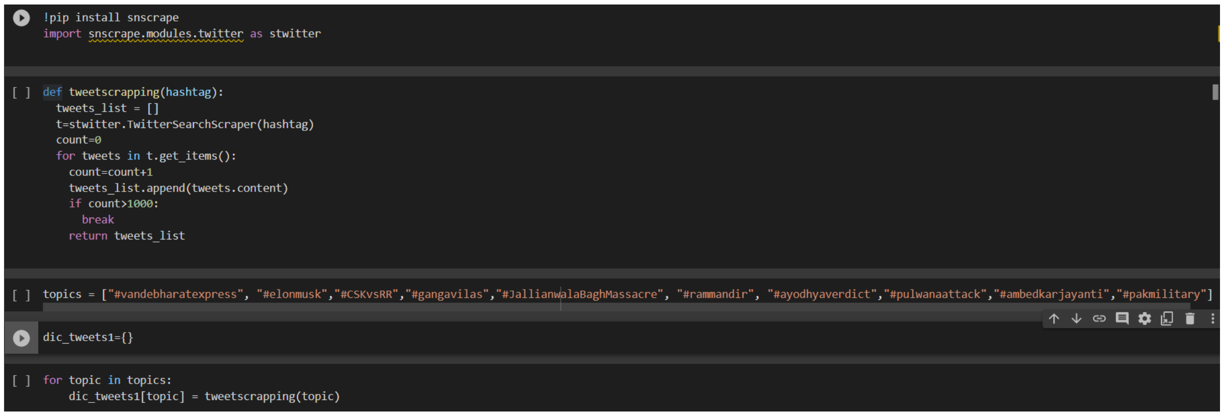
▼ Overall Approach :

A dataset of tweets from different topics is generated and after various preprocessing steps , **Pegasus** model is used as base model and fine tuned on our corpus to generate summaries. We considered **ROUGE** score to evaluate our model .

▼ Dataset Preparation :

- Using Snscrape(Python Library)

  We used Snscrape Python library to generate our dataset of tweets from 10 different trending topics.



```
!pip install snscrape
import snscrape.modules.twitter as stwitter
```

```
def tweetscrapping(hashtag):
  tweets_list = []
  t=stwitter.TwitterSearchScraper(hashtag)
  count=0
  for tweets in t.get_items():
    count=count+1
    tweets_list.append(tweets.content)
    if count>1000:
      break
    return tweets_list
```

```
topics = ["#vandebharatexpress", "#elonmusk","#CSKvsRR","#gangavilas","#JallianwalaBaghMassacre", "#rammandir", "#ayodhyaverdict","#pulwanaattack","#ambedkarjayanti","#pakmilitary"]
```

```
dic_tweets1={}
```

```
for topic in topics:
    dic_tweets1[topic] = tweetscrapping(topic)
```

- Using available automated platform Apify (https://console.apify.com/) to scrape tweets of a particular hashtag .

Generated dataset - **https://github.com/Srivastava-Rani-Aakanksha/NLP-Project-Abstractive-Text-Summarization-/blob/main/Dataset.csv**

Some other available datasets -

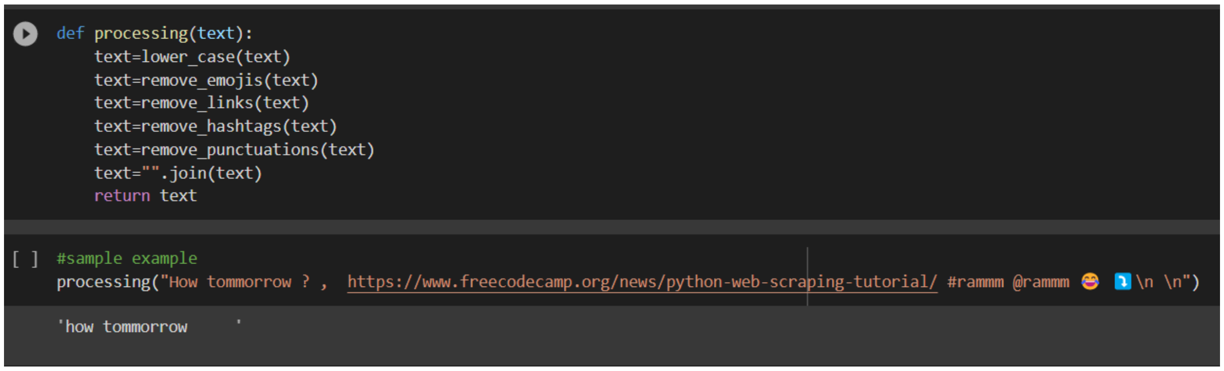
1. **https://github.com/kavgan/opinosis-summarization** (Graph algorithm based summarization framework
2. **https://github.com/guyfe/Tweetsumm** (A dataset focusing on summarization of dialogs, which represents the rich domain of Twitter customer care conversations and many more.)

▼ Preprocessing of Dataset :

Since we had created the corpus so lots of data cleaning and preprocessing was done before applying our model. We had only selected tweets of English language.

▼ Data Cleaning involves below steps:

1. Corpus Analysis
2. Lower case conversion of all tweets
3. Removal of all URLs
4. Removal of hashtags and mentions
5. Removal of emojis
6. Removal of punctuations
7. Removal of repeated tweets (redundancy removed)



```
def processing(text):
    text=lower_case(text)
    text=remove_emojis(text)
    text=remove_links(text)
    text=remove_hashtags(text)
    text=remove_punctuations(text)
    text="".join(text)
    return text
```

```
[ ] #sample example
    processing("How tommorrow ? ,  https://www.freecodecamp.org/news/python-web-scraping-tutorial/ #rammm @rammm 😂 ⤵️\n \n")

    'how tommorrow      '
```

Lexical Preprocessing - In NLP lexical processing refers to the process of analyzing words in a text. We had used it to transform the raw , unstructured text data into structured data which we had analyzed further .
It includes - Tokenization , Lemmatization , Stemming , Part of Speech Tagging (POS) ,Word sense disambiguation, Word Embeddings etc.

Tokenization - Using Python's NLTK library we had tokenized the final cleaned and preprocessed corpus .Basically tokenization means to break sentences (tweets) into further smaller units called tokens . Here we had done word tokenization.

For example, **"He is crying"** → **'He', 'is', 'crying'** .

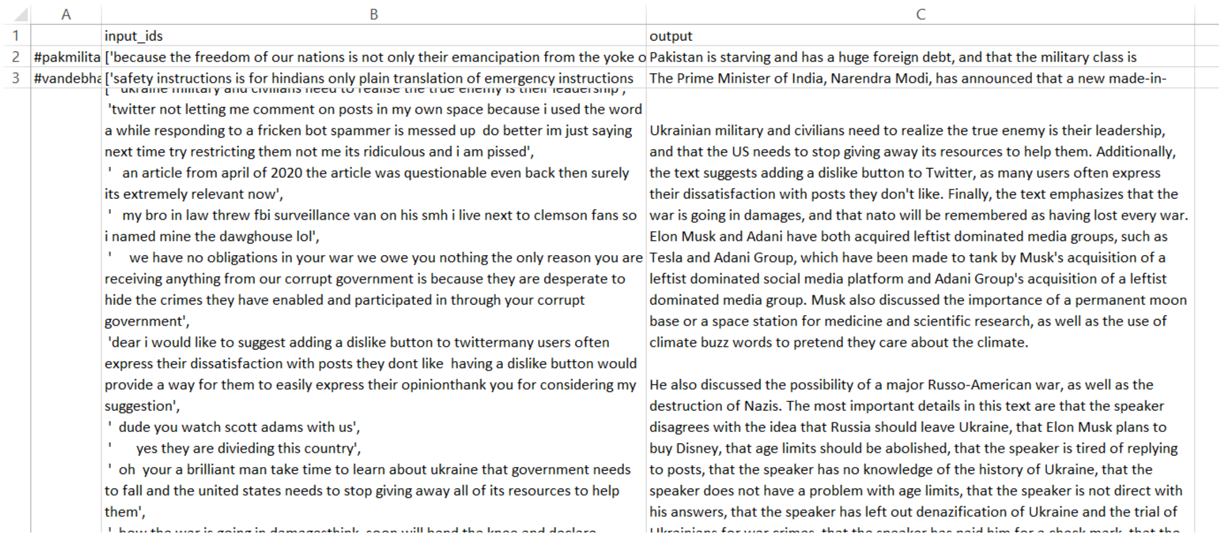
Lemmatization - Used it for reducing a word to its base or dictionary form, known as the lemma. It involves identifying the root form of a word while taking into account its part of speech (POS) tag.

For example, **"running"** → **"run"**, **"mice"** → **"mouse"**.

▼ Labelling the dataset by Summary Generation :

There are multiple way to do this -

1. Using Open AI GPT3 Model (But Open AI has a limit of tokens that it can summarize)
2. Manual Summary generation using tools like QuillBot .



	A	B	C
1		input_ids	output
2	#pakmilita	['because the freedom of our nations is not only their emancipation from the yoke o	Pakistan is starving and has a huge foreign debt, and that the military class is
3	#vandebha	['safety instructions is for hindians only plain translation of emergency instructions	The Prime Minister of India, Narendra Modi, has announced that a new made-in-
		['ukraine military and civilians need to realise the true enemy is their leadership', 'twitter not letting me comment on posts in my own space because i used the word a while responding to a fricken bot spammer is messed up do better im just saying next time try restricting them not me its ridiculous and i am pissed', ' an article from april of 2020 the article was questionable even back then surely its extremely relevant now', ' my bro in law threw fbi surveillance van on his smh i live next to clemson fans so i named mine the dawghouse lol', ' we have no obligations in your war we owe you nothing the only reason you are receiving anything from our corrupt government is because they are desperate to hide the crimes they have enabled and participated in through your corrupt government', 'dear i would like to suggest adding a dislike button to twittermany users often express their dissatisfaction with posts they dont like having a dislike button would provide a way for them to easily express their opinionthank you for considering my suggestion', ' dude you watch scott adams with us', ' yes they are divieding this country', ' oh your a brilliant man take time to learn about ukraine that government needs to fall and the united states needs to stop giving away all of its resources to help them', ' how the war is going in damagesthink soon will bend the knee and declare	Ukrainian military and civilians need to realize the true enemy is their leadership, and that the US needs to stop giving away its resources to help them. Additionally, the text suggests adding a dislike button to Twitter, as many users often express their dissatisfaction with posts they don't like. Finally, the text emphasizes that the war is going in damages, and that nato will be remembered as having lost every war. Elon Musk and Adani have both acquired leftist dominated media groups, such as Tesla and Adani Group, which have been made to tank by Musk's acquisition of a leftist dominated social media platform and Adani Group's acquisition of a leftist dominated media group. Musk also discussed the importance of a permanent moon base or a space station for medicine and scientific research, as well as the use of climate buzz words to pretend they care about the climate. He also discussed the possibility of a major Russo-American war, as well as the destruction of Nazis. The most important details in this text are that the speaker disagrees with the idea that Russia should leave Ukraine, that Elon Musk plans to buy Disney, that age limits should be abolished, that the speaker is tired of replying to posts, that the speaker has no knowledge of the history of Ukraine, that the speaker does not have a problem with age limits, that the speaker is not direct with his answers, that the speaker has left out denazification of Ukraine and the trial of Ukrainians for war crimes, that the speaker has paid him for a check mark, that the

Snippet from final dataset with hashtags, input id as list of tweets and output as manually generated summary.

▼ Training : Fine Tuning the pre trained Pegasus Model

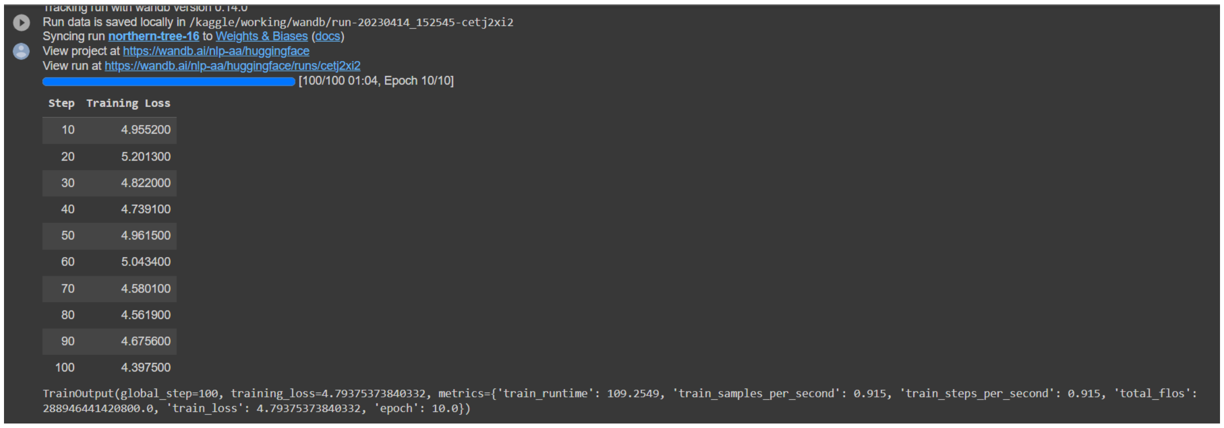
It is developed by Google AI Language. It is a sequence-to-sequence transformer model. It is trained on a large corpus of news articles and was fine-tuned on the CNN/Daily Mail dataset, which is a popular benchmark for text summarization.
As a seq-to-seq model, Pegasus consists of an encoder that processes the input text and a decoder that generates the output summary.

Given below are the steps involved in building and fine tuning of our model using Pegasus Model as base model.

- Imported Googles' Pegasus-large Model to use it as base model.
- Then above model was trained on our generated dataset.
- The developed model then stored on hugging face hub so as to use it further.
- Below are some parameters set during model training-

```
num_train_epochs=10,
per_device_train_batch_size=1,
```

  It takes more than an hour to train the pre trained model on our dataset and fine tuning it.
- Stored model on hugging face is pulled every time to test on test dataset.



```
Tracking run with wandb version 0.14.0
Run data is saved locally in /kaggle/working/wandb/run-20230414_152545-cetj2xi2
Syncing run northern-tree-16 to Weights & Biases (docs)
View project at https://wandb.ai/nlp-aa/huggingface
View run at https://wandb.ai/nlp-aa/huggingface/runs/cetj2xi2
[100/100 01:04, Epoch 10/10]

Step  Training Loss
  10       4.955200
  20       5.201300
  30       4.822000
  40       4.739100
  50       4.961500
  60       5.043400
  70       4.580100
  80       4.561900
  90       4.675600
 100       4.397500

TrainOutput(global_step=100, training_loss=4.79375373840332, metrics={'train_runtime': 109.2549, 'train_samples_per_second': 0.915, 'train_steps_per_second': 0.915, 'total_flos':
288946441420800.0, 'train_loss': 4.79375373840332, 'epoch': 10.0})
```

▼ Evaluation Metric :

There are several Evaluation Metric present like BLEU , ROUGE , METEOR. And we had used **ROUGE** metric for evaluation .It measures the overlap between the output and the reference summaries in terms of n-gram co-occurrences and word order.

Different ROUGE metrics are present like ROUGE1(it measures overlap of unigram), ROUGE2(it measures overlap of bigrams), ROUGEL(measures longest common subsequence).

Example -

Machine Generated Summary - I really loved eating Mangoes.

Human Reference Summary - I loved eating Mangoes.

Recall = No. of word matches / No. of words in reference

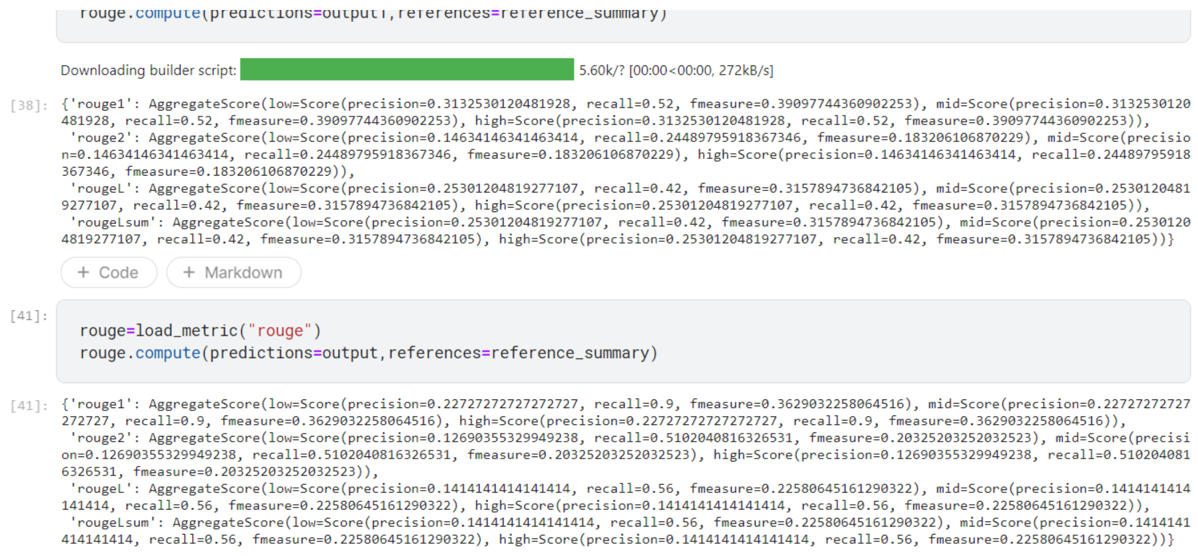
Precision =No. of word matches / No. of words in summary

F1- Score = 2*[(Precision * Recall) / (Precision + Recall)]

The highest ROUGE score achieved is 39%. Test dataset contains some tweets of two different trending hashtags.

ROUGE Score	Our Model	Base Pegasus Model
rouge-1	39%	36%
rouge-2	18.9%	20%
rouge-L	31%	22.8%

Below is code snippet from notebook



```
rouge.compute(predictions=output1,references=reference_summary)
```

Downloading builder script: 5.60k/? [00:00<00:00, 272kB/s]

```
[38]: {'rouge1': AggregateScore(low=Score(precision=0.3132530120481928, recall=0.52, fmeasure=0.39097744360902253), mid=Score(precision=0.3132530120481928, recall=0.52, fmeasure=0.39097744360902253), high=Score(precision=0.3132530120481928, recall=0.52, fmeasure=0.39097744360902253)),
 'rouge2': AggregateScore(low=Score(precision=0.14634146341463414, recall=0.24489795918367346, fmeasure=0.183206106870229), mid=Score(precision=0.14634146341463414, recall=0.24489795918367346, fmeasure=0.183206106870229), high=Score(precision=0.14634146341463414, recall=0.24489795918367346, fmeasure=0.183206106870229)),
 'rougeL': AggregateScore(low=Score(precision=0.25301204819277107, recall=0.42, fmeasure=0.3157894736842105), mid=Score(precision=0.25301204819277107, recall=0.42, fmeasure=0.3157894736842105), high=Score(precision=0.25301204819277107, recall=0.42, fmeasure=0.3157894736842105)),
 'rougeLsum': AggregateScore(low=Score(precision=0.25301204819277107, recall=0.42, fmeasure=0.3157894736842105), mid=Score(precision=0.25301204819277107, recall=0.42, fmeasure=0.3157894736842105), high=Score(precision=0.25301204819277107, recall=0.42, fmeasure=0.3157894736842105))}
```

```
[41]: rouge=load_metric("rouge")
      rouge.compute(predictions=output,references=reference_summary)
```

```
[41]: {'rouge1': AggregateScore(low=Score(precision=0.22727272727272727, recall=0.9, fmeasure=0.3629032258064516), mid=Score(precision=0.22727272727272727, recall=0.9, fmeasure=0.3629032258064516), high=Score(precision=0.22727272727272727, recall=0.9, fmeasure=0.3629032258064516)),
 'rouge2': AggregateScore(low=Score(precision=0.12690355329949238, recall=0.5102040816326531, fmeasure=0.20325203252032523), mid=Score(precision=0.12690355329949238, recall=0.5102040816326531, fmeasure=0.20325203252032523), high=Score(precision=0.12690355329949238, recall=0.5102040816326531, fmeasure=0.20325203252032523)),
 'rougeL': AggregateScore(low=Score(precision=0.1414141414141414, recall=0.56, fmeasure=0.22580645161290322), mid=Score(precision=0.1414141414141414, recall=0.56, fmeasure=0.22580645161290322), high=Score(precision=0.1414141414141414, recall=0.56, fmeasure=0.22580645161290322)),
 'rougeLsum': AggregateScore(low=Score(precision=0.1414141414141414, recall=0.56, fmeasure=0.22580645161290322), mid=Score(precision=0.1414141414141414, recall=0.56, fmeasure=0.22580645161290322), high=Score(precision=0.1414141414141414, recall=0.56, fmeasure=0.22580645161290322))}
```

▼ Key takeaways from Mandate :

- The Mandate task for generating tweets summary helped to explore more in the field of Natural Language Processing. We got to know how words , sentences are converted into embeddings for machines to understand them.
- We explored different pre trained models like BART , T5, Open- AI GPT3 , Pegasus and many more along with their features and drawbacks.
- We even concluded that still there is a lot to do in this NLP domain of summarization as it is challenging to capture the essence of input text in context with the sentence.

▼ Challenges faced :

1. Corpus generation - Since good twitter tweets datasets for summary generation are not available online so we have to frame our own datasets. Like for sentiment analysis we already have good online available datasets that we can use for it .
2. Large number of tweets. As we know to train the model for better accuracy we need large amount datasets but here for our project we had few amount of data to train the model.
3. Semantic Understanding - For a model to generate good summary it must understand the meaning of the input text and generate a summary that captures the essential information. This is a challenging task as it requires the model to understand the nuances of language, including the meanings of words, phrases, and sentences .
4. Also it is challenging to frame a summary that captures the most relevant information from all tweets.
5. As we know natural language is too ambiguous, many times same word, phrase holds different meaning depending on context. Capturing this ambiguity is quite challenging.
6. Evaluation Metrics - For extractive summarization we have good evaluation metrics but for abstractive summarization we don't have good metrics to capture the essence of generated summary.

▼ References :

1. Mandate Slides
2. https://huggingface.co/
3. https://github.com/sarahaman/CIS6930_TweetSum_Summarization/blob/main/model_finetuning/pegasus_model.ipyr
4. https://www.topcoder.com/thrive/articles/text-summarization-in-nlp
5. https://gist.github.com/jiahao87/50cec29725824da7ff6dd9314b53c4b3

You can refer below links for all notebooks, dataset and model -

- Collab Notebook Links

Notebook with all preprocessing steps that have been applied on the data to create corpus. (Basically building of datasets)

Google Colaboratory

https://colab.research.google.com/drive/1706-cb2XNM-why7mlFooGqJ9eWcCrje7?usp=sharing

Notebook with model training and fine tunning
https://colab.research.google.com/drive/19Aq9Vqit_1ghsqAuOjQjoWNpf9WXF4zv?usp=sharing
Notebook with ROUGE evaluation -https://colab.research.google.com/drive/1N17uWFcEjbk3i-R9nZg6nbsMsnPUCvdB?usp=sharing

- Git Hub Link -

https://github.com/Srivastava-Rani-Aakanksha/NLP-Project-Abstractive-Text-Summarization-.git

- Hugging Face Link -

Aakanksha1999 (Aakanksha Rani)
User profile of Aakanksha Rani on Hugging Face

https://huggingface.co/Aakanksha1999

Aakanksha1999

huggingface.co/Aakanksha1999