

✧ Importing Libraries

```
import warnings
warnings.filterwarnings("ignore")
```

```
!pip install datasets
```

```
Requirement already satisfied: datasets in /usr/local/lib/python3.10/dist-packages (2.18.0)
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-packages (from datasets) (3.13.4)
Requirement already satisfied: numpy<=1.17 in /usr/local/lib/python3.10/dist-packages (from datasets) (1.25.2)
Requirement already satisfied: pyarrow>=12.0.0 in /usr/local/lib/python3.10/dist-packages (from datasets) (14.0.2)
Requirement already satisfied: pyarrow-hotfix in /usr/local/lib/python3.10/dist-packages (from datasets) (0.6)
Requirement already satisfied: dill<0.3.9,>=0.3.0 in /usr/local/lib/python3.10/dist-packages (from datasets) (0.3.8)
Requirement already satisfied: pandas in /usr/local/lib/python3.10/dist-packages (from datasets) (2.0.3)
Requirement already satisfied: requests>=2.19.0 in /usr/local/lib/python3.10/dist-packages (from datasets) (2.31.0)
Requirement already satisfied: tqdm>=4.62.1 in /usr/local/lib/python3.10/dist-packages (from datasets) (4.66.2)
Requirement already satisfied: xxhash in /usr/local/lib/python3.10/dist-packages (from datasets) (3.4.1)
Requirement already satisfied: multiprocessing in /usr/local/lib/python3.10/dist-packages (from datasets) (0.70.16)
Requirement already satisfied: fsspec[http]<=2024.2.0,>=2023.1.0 in /usr/local/lib/python3.10/dist-packages (from datasets) (2024.2.0)
Requirement already satisfied: aiohttp in /usr/local/lib/python3.10/dist-packages (from datasets) (3.9.3)
Requirement already satisfied: huggingface-hub>=0.19.4 in /usr/local/lib/python3.10/dist-packages (from datasets) (0.20.3)
Requirement already satisfied: packaging in /usr/local/lib/python3.10/dist-packages (from datasets) (24.0)
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.10/dist-packages (from datasets) (6.0.1)
Requirement already satisfied: aiosignal>=1.1.2 in /usr/local/lib/python3.10/dist-packages (from aiohttp->datasets) (1.3.1)
Requirement already satisfied: attrs>=17.3.0 in /usr/local/lib/python3.10/dist-packages (from aiohttp->datasets) (23.2.0)
Requirement already satisfied: frozenlist>=1.1.1 in /usr/local/lib/python3.10/dist-packages (from aiohttp->datasets) (1.4.1)
Requirement already satisfied: multidict<7.0,>=4.5 in /usr/local/lib/python3.10/dist-packages (from aiohttp->datasets) (6.0.5)
Requirement already satisfied: yarl<2.0,>=1.0 in /usr/local/lib/python3.10/dist-packages (from aiohttp->datasets) (1.9.4)
Requirement already satisfied: async-timeout<5.0,>=4.0 in /usr/local/lib/python3.10/dist-packages (from aiohttp->datasets) (4.0.3)
Requirement already satisfied: typing-extensions>=3.7.4.3 in /usr/local/lib/python3.10/dist-packages (from huggingface-hub>=0.19.4->datasets) (4.11.0)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests>=2.19.0->datasets) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests>=2.19.0->datasets) (3.10.1)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests>=2.19.0->datasets) (2.2.3)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests>=2.19.0->datasets) (2024.7.4)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.10/dist-packages (from pandas->datasets) (2.9.0)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas->datasets) (2023.4)
Requirement already satisfied: tzdata>=2022.1 in /usr/local/lib/python3.10/dist-packages (from pandas->datasets) (2024.1)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.8.2->pandas->datasets) (1.16.0)
```

```
!pip install rouge
```

```
Requirement already satisfied: rouge in /usr/local/lib/python3.10/dist-packages (1.0.1)
Requirement already satisfied: six in /usr/local/lib/python3.10/dist-packages (from rouge) (1.16.0)
```

```
! pip install torch==1.8.2+cu111 torchvision==0.9.2+cu111 torchaudio===0.8.2 -f https://download.pytorch.org/whl/lts/1.8/torch_l
! pip install datasets
! pip install rouge
```

```
Looking in links: https://download.pytorch.org/whl/lts/1.8/torch\_lts.html
ERROR: Could not find a version that satisfies the requirement torch==1.8.2+cu111 (from versions: 1.11.0, 1.12.0, 1.12.1, 1.13.0)
ERROR: No matching distribution found for torch==1.8.2+cu111

Requirement already satisfied: datasets in /usr/local/lib/python3.10/dist-packages (2.18.0)
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-packages (from datasets) (3.13.4)
Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.10/dist-packages (from datasets) (1.25.2)
Requirement already satisfied: pyarrow>=12.0.0 in /usr/local/lib/python3.10/dist-packages (from datasets) (14.0.2)
Requirement already satisfied: pyarrow-hotfix in /usr/local/lib/python3.10/dist-packages (from datasets) (0.6)
Requirement already satisfied: dill<0.3.9,>=0.3.0 in /usr/local/lib/python3.10/dist-packages (from datasets) (0.3.8)
Requirement already satisfied: pandas in /usr/local/lib/python3.10/dist-packages (from datasets) (2.0.3)
Requirement already satisfied: requests>=2.19.0 in /usr/local/lib/python3.10/dist-packages (from datasets) (2.31.0)
Requirement already satisfied: tqdm>=4.62.1 in /usr/local/lib/python3.10/dist-packages (from datasets) (4.66.2)
Requirement already satisfied: xxhash in /usr/local/lib/python3.10/dist-packages (from datasets) (3.4.1)
Requirement already satisfied: multiprocessing in /usr/local/lib/python3.10/dist-packages (from datasets) (0.70.16)
Requirement already satisfied: fsspec[http]<=2024.2.0,>=2023.1.0 in /usr/local/lib/python3.10/dist-packages (from datasets) (2024.2.0)
Requirement already satisfied: aiohttp in /usr/local/lib/python3.10/dist-packages (from datasets) (3.9.3)
Requirement already satisfied: huggingface-hub>=0.19.4 in /usr/local/lib/python3.10/dist-packages (from datasets) (0.20.3)
Requirement already satisfied: packaging in /usr/local/lib/python3.10/dist-packages (from datasets) (24.0)
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.10/dist-packages (from datasets) (6.0.1)
Requirement already satisfied: aiosignal>=1.1.2 in /usr/local/lib/python3.10/dist-packages (from aiohttp->datasets) (1.3.1)
Requirement already satisfied: attrs>=17.3.0 in /usr/local/lib/python3.10/dist-packages (from aiohttp->datasets) (23.2.0)
Requirement already satisfied: frozenlist>=1.1.1 in /usr/local/lib/python3.10/dist-packages (from aiohttp->datasets) (1.4.1)
Requirement already satisfied: multidict<7.0,>=4.5 in /usr/local/lib/python3.10/dist-packages (from aiohttp->datasets) (6.0.5)
Requirement already satisfied: yarl<2.0,>=1.0 in /usr/local/lib/python3.10/dist-packages (from aiohttp->datasets) (1.9.4)
Requirement already satisfied: async-timeout<5.0,>=4.0 in /usr/local/lib/python3.10/dist-packages (from aiohttp->datasets) (4.0.3)
Requirement already satisfied: typing-extensions>=3.7.4.3 in /usr/local/lib/python3.10/dist-packages (from huggingface-hub>=0.19.4->datasets) (4.11.0)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests>=2.19.0->datasets) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests>=2.19.0->datasets) (3.10.1)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests>=2.19.0->datasets) (2.2.3)
```

```
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests>=2.19.0->dataset)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.10/dist-packages (from pandas->datasets) (2.
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas->datasets) (2023.4)
Requirement already satisfied: tzdata>=2022.1 in /usr/local/lib/python3.10/dist-packages (from pandas->datasets) (2024.1)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.8.2->pandas->dat
Requirement already satisfied: rouge in /usr/local/lib/python3.10/dist-packages (1.0.1)
Requirement already satisfied: six in /usr/local/lib/python3.10/dist-packages (from rouge) (1.16.0)
```

! pip install transformers

```
Requirement already satisfied: transformers in /usr/local/lib/python3.10/dist-packages (4.38.2)
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-packages (from transformers) (3.13.4)
Requirement already satisfied: huggingface-hub<1.0,=>0.19.3 in /usr/local/lib/python3.10/dist-packages (from transformers) (
Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.10/dist-packages (from transformers) (1.25.2)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from transformers) (24.0)
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.10/dist-packages (from transformers) (6.0.1)
Requirement already satisfied: regex!=2019.12.17 in /usr/local/lib/python3.10/dist-packages (from transformers) (2023.12.25)
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (from transformers) (2.31.0)
Requirement already satisfied: tokenizers<0.19,=>0.14 in /usr/local/lib/python3.10/dist-packages (from transformers) (0.15.2)
Requirement already satisfied: safetensors>=0.4.1 in /usr/local/lib/python3.10/dist-packages (from transformers) (0.4.2)
Requirement already satisfied: tqdm>=4.27 in /usr/local/lib/python3.10/dist-packages (from transformers) (4.66.2)
Requirement already satisfied: fsspec>=2023.5.0 in /usr/local/lib/python3.10/dist-packages (from huggingface-hub<1.0,=>0.19.
Requirement already satisfied: typing-extensions>=3.7.4.3 in /usr/local/lib/python3.10/dist-packages (from huggingface-hub<1
Requirement already satisfied: charset-normalizer<4,=>2 in /usr/local/lib/python3.10/dist-packages (from requests->transform
Requirement already satisfied: idna<4,=>2.5 in /usr/local/lib/python3.10/dist-packages (from requests->transformers) (3.6)
Requirement already satisfied: urllib3<3,=>1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests->transformers) (
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests->transformers) (
```

!pip install transformers[torch]



```
Requirement already satisfied: transformers[torch] in /usr/local/lib/python3.10/dist-packages (4.38.2)
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-packages (from transformers[torch]) (3.13.4)
Requirement already satisfied: huggingface-hub<1.0,=>0.19.3 in /usr/local/lib/python3.10/dist-packages (from transformers[to
Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.10/dist-packages (from transformers[torch]) (1.25.2)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from transformers[torch]) (24.0)
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.10/dist-packages (from transformers[torch]) (6.0.1)
Requirement already satisfied: regex!=2019.12.17 in /usr/local/lib/python3.10/dist-packages (from transformers[torch]) (2023
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (from transformers[torch]) (2.31.0)
Requirement already satisfied: tokenizers<0.19,=>0.14 in /usr/local/lib/python3.10/dist-packages (from transformers[torch])
Requirement already satisfied: safetensors>=0.4.1 in /usr/local/lib/python3.10/dist-packages (from transformers[torch]) (0.4
Requirement already satisfied: tqdm>=4.27 in /usr/local/lib/python3.10/dist-packages (from transformers[torch]) (4.66.2)
Requirement already satisfied: torch in /usr/local/lib/python3.10/dist-packages (from transformers[torch]) (2.2.1+cu121)
Requirement already satisfied: accelerate>=0.21.0 in /usr/local/lib/python3.10/dist-packages (from transformers[torch]) (0.2
Requirement already satisfied: psutil in /usr/local/lib/python3.10/dist-packages (from accelerate>=0.21.0->transformers[torc
Requirement already satisfied: fsspec>=2023.5.0 in /usr/local/lib/python3.10/dist-packages (from huggingface-hub<1.0,=>0.19.
Requirement already satisfied: typing-extensions>=3.7.4.3 in /usr/local/lib/python3.10/dist-packages (from huggingface-hub<1
Requirement already satisfied: sympy in /usr/local/lib/python3.10/dist-packages (from torch->transformers[torch]) (1.12)
Requirement already satisfied: networkx in /usr/local/lib/python3.10/dist-packages (from torch->transformers[torch]) (3.3)
Requirement already satisfied: Jinja2 in /usr/local/lib/python3.10/dist-packages (from torch->transformers[torch]) (3.1.3)
Requirement already satisfied: nvidia-cuda-nvrtc-cu12==12.1.105 in /usr/local/lib/python3.10/dist-packages (from torch->tran
Requirement already satisfied: nvidia-cuda-runtime-cu12==12.1.105 in /usr/local/lib/python3.10/dist-packages (from torch->tr
Requirement already satisfied: nvidia-cuda-cupti-cu12==12.1.105 in /usr/local/lib/python3.10/dist-packages (from torch->tran
Requirement already satisfied: nvidia-cudnn-cu12==8.9.2.26 in /usr/local/lib/python3.10/dist-packages (from torch->transform
Requirement already satisfied: nvidia-cublas-cu12==12.1.3.1 in /usr/local/lib/python3.10/dist-packages (from torch->transform
Requirement already satisfied: nvidia-cufft-cu12==11.0.2.54 in /usr/local/lib/python3.10/dist-packages (from torch->transform
Requirement already satisfied: nvidia-curand-cu12==10.3.2.106 in /usr/local/lib/python3.10/dist-packages (from torch->transf
Requirement already satisfied: nvidia-cusolver-cu12==11.4.5.107 in /usr/local/lib/python3.10/dist-packages (from torch->tran
Requirement already satisfied: nvidia-cusparse-cu12==12.1.0.106 in /usr/local/lib/python3.10/dist-packages (from torch->tran
Requirement already satisfied: nvidia-nccl-cu12==2.19.3 in /usr/local/lib/python3.10/dist-packages (from torch->transformers
Requirement already satisfied: nvidia-nvtx-cu12==12.1.105 in /usr/local/lib/python3.10/dist-packages (from torch->transforme
Requirement already satisfied: triton==2.2.0 in /usr/local/lib/python3.10/dist-packages (from torch->transformers[torch]) (2
Requirement already satisfied: nvidia-nvjitlink-cu12 in /usr/local/lib/python3.10/dist-packages (from nvidia-cusolver-cu12==
Requirement already satisfied: charset-normalizer<4,=>2 in /usr/local/lib/python3.10/dist-packages (from requests->transform
Requirement already satisfied: idna<4,=>2.5 in /usr/local/lib/python3.10/dist-packages (from requests->transformers[torch])
Requirement already satisfied: urllib3<3,=>1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests->transformers[to
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests->transformers[to
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.10/dist-packages (from Jinja2->torch->transformers[
Requirement already satisfied: mpmath>=0.19 in /usr/local/lib/python3.10/dist-packages (from sympy->torch->transformers[torc
```

!pip install accelerate

```
Requirement already satisfied: accelerate in /usr/local/lib/python3.10/dist-packages (0.29.2)
Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.10/dist-packages (from accelerate) (1.25.2)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from accelerate) (24.0)
Requirement already satisfied: psutil in /usr/local/lib/python3.10/dist-packages (from accelerate) (5.9.5)
Requirement already satisfied: pyyaml in /usr/local/lib/python3.10/dist-packages (from accelerate) (6.0.1)
Requirement already satisfied: torch>=1.10.0 in /usr/local/lib/python3.10/dist-packages (from accelerate) (2.2.1+cu121)
Requirement already satisfied: huggingface-hub in /usr/local/lib/python3.10/dist-packages (from accelerate) (0.20.3)
Requirement already satisfied: safetensors>=0.3.1 in /usr/local/lib/python3.10/dist-packages (from accelerate) (0.4.2)
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-packages (from torch>=1.10.0->accelerate) (3.13.4)
Requirement already satisfied: typing-extensions>=4.8.0 in /usr/local/lib/python3.10/dist-packages (from torch>=1.10.0->acce
```

```
Requirement already satisfied: sympy in /usr/local/lib/python3.10/dist-packages (from torch>=1.10.0->accelerate) (1.12)
Requirement already satisfied: networkx in /usr/local/lib/python3.10/dist-packages (from torch>=1.10.0->accelerate) (3.3)
Requirement already satisfied: jinja2 in /usr/local/lib/python3.10/dist-packages (from torch>=1.10.0->accelerate) (3.1.3)
Requirement already satisfied: fsspec in /usr/local/lib/python3.10/dist-packages (from torch>=1.10.0->accelerate) (2023.6.0)
Requirement already satisfied: nvidia-cuda-nvrtc-cu12==12.1.105 in /usr/local/lib/python3.10/dist-packages (from torch>=1.10.0->accelerate) (12.1.105)
Requirement already satisfied: nvidia-cuda-runtime-cu12==12.1.105 in /usr/local/lib/python3.10/dist-packages (from torch>=1.10.0->accelerate) (12.1.105)
Requirement already satisfied: nvidia-cuda-cupti-cu12==12.1.105 in /usr/local/lib/python3.10/dist-packages (from torch>=1.10.0->accelerate) (12.1.105)
Requirement already satisfied: nvidia-cudnn-cu12==8.9.2.26 in /usr/local/lib/python3.10/dist-packages (from torch>=1.10.0->accelerate) (8.9.2.26)
Requirement already satisfied: nvidia-cublas-cu12==12.1.3.1 in /usr/local/lib/python3.10/dist-packages (from torch>=1.10.0->accelerate) (12.1.3.1)
Requirement already satisfied: nvidia-cufft-cu12==11.0.2.54 in /usr/local/lib/python3.10/dist-packages (from torch>=1.10.0->accelerate) (11.0.2.54)
Requirement already satisfied: nvidia-curand-cu12==10.3.2.106 in /usr/local/lib/python3.10/dist-packages (from torch>=1.10.0->accelerate) (10.3.2.106)
Requirement already satisfied: nvidia-cusolver-cu12==11.4.5.107 in /usr/local/lib/python3.10/dist-packages (from torch>=1.10.0->accelerate) (11.4.5.107)
Requirement already satisfied: nvidia-cusparse-cu12==12.1.0.106 in /usr/local/lib/python3.10/dist-packages (from torch>=1.10.0->accelerate) (12.1.0.106)
Requirement already satisfied: nvidia-nccl-cu12==2.19.3 in /usr/local/lib/python3.10/dist-packages (from torch>=1.10.0->accelerate) (2.19.3)
Requirement already satisfied: nvidia-nvtx-cu12==12.1.105 in /usr/local/lib/python3.10/dist-packages (from torch>=1.10.0->accelerate) (12.1.105)
Requirement already satisfied: triton==2.2.0 in /usr/local/lib/python3.10/dist-packages (from torch>=1.10.0->accelerate) (2.2.0)
Requirement already satisfied: nvidia-nvjitlink-cu12 in /usr/local/lib/python3.10/dist-packages (from torch>=1.10.0->accelerate) (12.1.105)
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (from huggingface-hub->accelerate) (2.31.0)
Requirement already satisfied: tqdm>=4.42.1 in /usr/local/lib/python3.10/dist-packages (from huggingface-hub->accelerate) (4.64.0)
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.10/dist-packages (from jinja2->torch>=1.10.0->accelerate) (2.1.2)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests->huggingface-hub) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests->huggingface-hub) (3.6)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests->huggingface-hub) (2.0.7)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests->huggingface-hub) (2023.7.22)
Requirement already satisfied: mpmath>=0.19 in /usr/local/lib/python3.10/dist-packages (from sympy->torch>=1.10.0->accelerate) (3.1.0)
```

```
from google.colab import drive
from datasets import load_dataset

import nltk
import string
from collections import Counter
from nltk.corpus import stopwords
nltk.download('stopwords')
nltk.download('punkt')
import pandas as pd
import numpy as np
import re
from tqdm import tqdm
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split

from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.layers import Input, LSTM, Embedding, Dense, \
    Concatenate, TimeDistributed
from tensorflow.keras.models import Model
from tensorflow.keras.callbacks import EarlyStopping
from matplotlib import pyplot
from rouge import Rouge
from bs4 import BeautifulSoup
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data] Package stopwords is already up-to-date!
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data] Package punkt is already up-to-date!
```

✓ Loading Data

```
# Connecting to drive to store the models
drive.mount('/content/drive')
```

```
# Loading the data set
dataset = load_dataset("multi_news")
df = dataset['train'].to_pandas()
```

```
df.rename(columns={'document': 'text'}, inplace=True)
```

```
df.head(2)
```

	text	summary
0	National Archives \n \n Yes, it's that time ag... — The unemployment rate dropped to 8.2% last m...	
1	LOS ANGELES (AP) — In her first interview sinc... — Shelly Sterling plans "eventually" to divorc...	

Next steps: [Generate code with df](#) [View recommended plots](#)

✓ Data Pre-Processing

df.info() #Getting the info

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 44972 entries, 0 to 44971
Data columns (total 2 columns):
 #   Column      Non-Null Count  Dtype
---  ---
 0   text        44972 non-null  object
 1   summary     44972 non-null  object
dtypes: object(2)
memory usage: 702.8+ KB
```

df.isnull().sum() #Checking the null records

```
text      0
summary   0
dtype: int64
```

df.duplicated().sum() #Checking the duplicate records

```
0
```

```
contraction_mapping = {"ain't": "is not", "aren't": "are not", "can't": "cannot", "'cause": "because", "could've": "could have",
                        "didn't": "did not", "doesn't": "does not", "don't": "do not", "hadn't": "had not", "hasn't": "has r",
                        "he'd": "he would", "he'll": "he will", "he's": "he is", "how'd": "how did", "how'd'y": "how do you",
                        "I'd": "I would", "I'd've": "I would have", "I'll": "I will", "I'll've": "I will have", "I'm": "I am",
                        "i'd've": "i would have", "i'll": "i will", "i'll've": "i will have", "i'm": "i am", "i've": "i have",
                        "it'd've": "it would have", "it'll": "it will", "it'll've": "it will have", "it's": "it is", "let's":
                        "mayn't": "may not", "might've": "might have", "mightn't": "might not", "mightn't've": "might not have",
                        "mustn't": "must not", "mustn't've": "must not have", "needn't": "need not", "needn't've": "need not",
                        "oughtn't": "ought not", "oughtn't've": "ought not have", "shan't": "shall not", "sha'n't": "shall no",
                        "she'd": "she would", "she'd've": "she would have", "she'll": "she will", "she'll've": "she will have",
                        "should've": "should have", "shouldn't": "should not", "shouldn't've": "should not have", "so've": "s",
                        "this's": "this is", "that'd": "that would", "that'd've": "that would have", "that's": "that is", "the",
                        "there'd've": "there would have", "there's": "there is", "here's": "here is", "they'd": "they would",
                        "they'll": "they will", "they'll've": "they will have", "they're": "they are", "they've": "they have",
                        "wasn't": "was not", "we'd": "we would", "we'd've": "we would have", "we'll": "we will", "we'll've":
                        "we've": "we have", "weren't": "were not", "what'll": "what will", "what'll've": "what will have", "w",
                        "what's": "what is", "what've": "what have", "when's": "when is", "when've": "when have", "where'd":
                        "where've": "where have", "who'll": "who will", "who'll've": "who will have", "who's": "who is", "whc",
                        "why's": "why is", "why've": "why have", "will've": "will have", "won't": "will not", "won't've": "wi",
                        "would've": "would have", "wouldn't": "would not", "wouldn't've": "would not have", "y'all": "you all",
                        "y'all'd": "you all would", "y'all'd've": "you all would have", "y'all're": "you all are", "y'all've": "
                        "you'd": "you would", "you'd've": "you would have", "you'll": "you will", "you'll've": "you will have",
                        "you're": "you are", "you've": "you have"}
```

```
def text_strip(s):
    """
    This function is used to remove the punctuations, special characters, and stopwords from the text.
    """
    # Convert the text to lowercase
    s = s.lower()

    # Removing punctuations using the translate method
    s = s.translate(str.maketrans('', '', string.punctuation))

    # Replacing special characters with spaces using regular expressions
    s = re.sub(r"[<>()|&@#%*\[\]\{\}\|'\";?~*!]", ' ', str(s))

    # Replacing multiple spaces with a single space
    s = re.sub("(\.\s+)", ' ', str(s))
    s = re.sub("(\-\s+)", ' ', str(s))
    s = re.sub("(\:\s+)", ' ', str(s))

    # Expand contractions using a predefined mapping
    s = ' '.join([contraction_mapping[t] if t in contraction_mapping else t for t in s.split(" ")])

    # Removing punctuations again after expanding contractions
    s = s.translate(str.maketrans('', '', string.punctuation))

    # Removing stopwords
    stop_words = set(stopwords.words('english'))
    return ' '.join(word for word in s.split() if word not in stop_words)
```



```
text = []
for t in df['text']:
    text.append(text_strip(t))

summary = []
for t in df['summary']:
    summary.append(text_strip(t))
```

```
# Adding the START and END to the summaries
summary = ['_START_' + str(t) + '_END_' for t in summary]
```

```
df['cleaned_text'] = pd.Series(text)
df['cleaned_summary'] = pd.Series(summary)
```

```
df.head(2)
```

	text	summary	cleaned_text	cleaned_summary	
0	National Archives Yes, it's that time ag...	The unemployment rate dropped to 8.2% last m...	national archives yes it's time folks it's fir...	_START_ - unemployment rate dropped 82 last mo...	 

Next steps: [Generate code with df](#) [View recommended plots](#)

```
df = df.dropna(axis=0)
```

```
# This code will display the distribution of lengths on text and summaries
text_count = []
summary_count = []

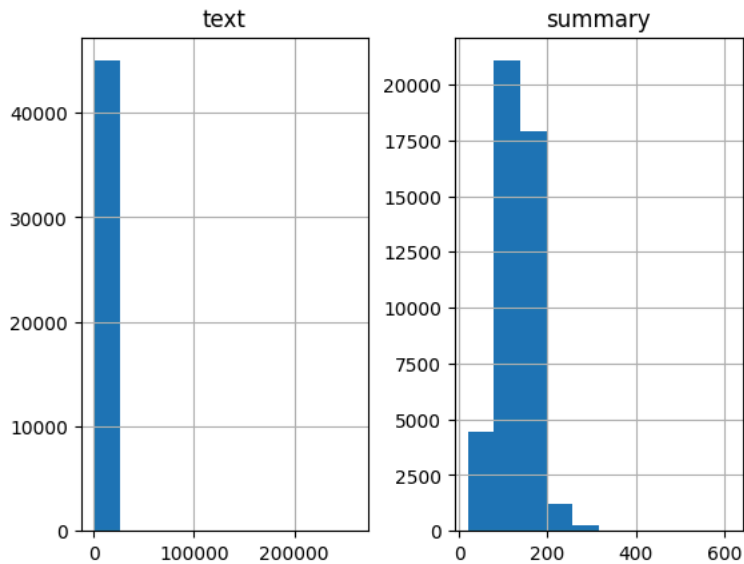
for sent in df['cleaned_text']:
    text_count.append(len(sent.split()))

for sent in df['cleaned_summary']:
    summary_count.append(len(sent.split()))

graph_df = pd.DataFrame()

graph_df['text'] = text_count
graph_df['summary'] = summary_count

graph_df.hist(bins = 10)
plt.show()
```



As we can see, we have outliers in the text. So, we are setting boundaries on the text and summary lengths to sample the dataset as the data set is huge

```
#Check how many records of summary have 128 words
count=0
for i in df['cleaned_summary']:
    if(len(i.split())<=128):
        count=count+1
print(count)
```

21081

```
# Check how many records of text have 1024 words
cnt = 0
for i in df['cleaned_text']:
    if len(i.split()) <= 1024:
        cnt = cnt + 1
print(cnt)
```

29376

```
# Model to summarize the text between 0-1024 words for text and 0-128 words for summary
max_text_len = 1024
max_summary_len = 128
```

```
# Selecting the text and summaries which fall below max length
cleaned_text = np.array(df['cleaned_text'])
cleaned_summary= np.array(df['cleaned_summary'])

short_text = []
short_summary = []

for i in range(len(cleaned_text)):
    if len(cleaned_summary[i].split()) <= max_summary_len and len(cleaned_text[i].split()) <= max_text_len:
        short_text.append(cleaned_text[i])
        short_summary.append(cleaned_summary[i])

filtered_df = pd.DataFrame({'text': short_text, 'summary': short_summary})
```

```
filtered_df.head()
```

	text	summary	
0	national archives yes it's time folks it's fir...	_START_ – unemployment rate dropped 82 last mo...	
1	los angeles ap — first interview since nba ban...	_START_ – shelly sterling plans eventually div...	
2	gaithersburg md ap — small private jet crashed...	_START_ – twinengine embraer jet faa describes...	
3	tucker carlson exposes sexism twitter	_START_ – tucker carlson deep doodoo	

Next steps: [Generate code with filtered_df](#) [View recommended plots](#)

```
filtered_df.shape
```

```
(16604, 2)
```

We noticed that some summaries are larger compare to texts. This summaries process may include redundant information.

```
# This will check the count of summaries that are larger compare to text
count = 0
for index, row in filtered_df.iterrows():
    if len(row['text'].split()) < len(row['summary'].split()):
        count += 1

print(count)
```

```
238
```

```
# Removing the records where summary length is larger than the text
filtered_df = filtered_df[filtered_df.apply(lambda row: len(row['text'].split()) > len(row['summary'].split()), axis=1)]
```

```
filtered_df.shape
```

```
(16357, 2)
```

```
# Adding sostok and eostok
```

```
filtered_df['summary'] = filtered_df['summary'].apply(lambda x: 'sostok ' + x + ' eostok')
filtered_df.head(2)
```

	text	summary	
0	national archives yes it's time folks it's fir...	sostok _START_ – unemployment rate dropped 82 ...	
1	los angeles ap — first interview since nba	sostok _START_ – shelly sterling plans	

Next steps: [Generate code with filtered_df](#) [View recommended plots](#)

```
# Splitting the data into train and validation
x_train, x_val, y_train, y_val = train_test_split(
    np.array(filtered_df["text"]),
    np.array(filtered_df["summary"]),
    test_size=0.2,
    random_state=0,
    shuffle=True,
)
```

✓ Preparing Tokenizers

```
# Preparing a tokenizer on text data
x_tokenizer = Tokenizer()
x_tokenizer.fit_on_texts(list(x_train))
```

```
# Getting the counts and percentage of rare words
# Threshold is set to 5, means words that are repeated less than 5 time are considered rare
thresh = 5

cnt = 0
tot_cnt = 0

for key, value in x_tokenizer.word_counts.items():
    tot_cnt = tot_cnt + 1
    if value < thresh:
        cnt = cnt + 1

print("Size of the vocabulary: ", tot_cnt)
print("Count of rare words: ", cnt)
print("% of rare words in vocabulary: ", (cnt / tot_cnt) * 100)
print("Count of most common words: ", tot_cnt-cnt)

Size of the vocabulary: 218724
Count of rare words: 161506
% of rare words in vocabulary: 73.84009070792415
Count of most common words: 57218
```

```
# Preparing a tokenizer
# Not considering the rare words
x_tokenizer = Tokenizer(num_words = tot_cnt - cnt)
x_tokenizer.fit_on_texts(list(x_train))

# Converting text sequences to integer sequences
x_tr_seq = x_tokenizer.texts_to_sequences(x_train)
x_val_seq = x_tokenizer.texts_to_sequences(x_val)

# Padding zero's upto maximum length
x_tr = pad_sequences(x_tr_seq, maxlen=max_text_len, padding='post')
x_val = pad_sequences(x_val_seq, maxlen=max_text_len, padding='post')

x_voc = x_tokenizer.num_words + 1

print("Size of vocabulary in X = {}".format(x_voc))

Size of vocabulary in X = 57219
```

```
# Preparing a tokenizer on summary data
y_tokenizer = Tokenizer()
y_tokenizer.fit_on_texts(list(y_train))

thresh = 5
cnt = 0
tot_cnt = 0

for key, value in y_tokenizer.word_counts.items():
    tot_cnt = tot_cnt + 1
    if value < thresh:
        cnt = cnt + 1

print("Size of the vocabulary: ", tot_cnt)
print("Count of rare words: ", cnt)
print("% of rare words in vocabulary: ", (cnt / tot_cnt) * 100)
print("Count of most common words: ", tot_cnt-cnt)

# Preparing a tokenizer
# not considering the rare words
y_tokenizer = Tokenizer(num_words=tot_cnt-cnt)
y_tokenizer.fit_on_texts(list(y_train))

# Converting text sequences to integer sequences
y_tr_seq = y_tokenizer.texts_to_sequences(y_train)
y_val_seq = y_tokenizer.texts_to_sequences(y_val)

# Padding zero's upto maximum length
y_tr = pad_sequences(y_tr_seq, maxlen=max_summary_len, padding='post')
y_val = pad_sequences(y_val_seq, maxlen=max_summary_len, padding='post')

y_voc = y_tokenizer.num_words + 1

print("Size of vocabulary in Y = {}".format(y_voc))
```



```

Size of the vocabulary: 77611
Count of rare words: 57592
% of rare words in vocabulary: 74.2059759570164
Count of most common words: 20019
Size of vocabulary in Y = 20020

```

```

ind = []

# Iterating over each sample in the dataset
for i in range(len(y_tr)):
    cnt = 0
    # Counting the number of non-zero tokens in the summary
    for j in y_tr[i]:
        if j != 0:
            cnt = cnt + 1
    # If the count is 2 (indicating only 'START' and 'END' tokens are present), adding the index to the list
    if cnt == 2:
        ind.append(i)

# Removing samples with empty summaries from both x_tr and y_tr
y_tr = np.delete(y_tr, ind, axis=0)
x_tr = np.delete(x_tr, ind, axis=0)

```

```

ind = []
for i in range(len(y_val)):
    cnt = 0
    for j in y_val[i]:
        if j != 0:
            cnt = cnt + 1
    if cnt == 2:
        ind.append(i)

y_val = np.delete(y_val, ind, axis=0)
x_val = np.delete(x_val, ind, axis=0)

```

✓ Seq2Seq Model with LSTM

```

latent_dim = 300 # Dimensionality of the latent space
embedding_dim = 200 # Dimensionality of the word embeddings

# Encoder
encoder_inputs = Input(shape=(max_text_len, ))

# Embedding layer for input sequences
enc_emb = Embedding(x_voc, embedding_dim, trainable=True)(encoder_inputs)

# First LSTM layer of the encoder
encoder_lstm1 = LSTM(latent_dim, return_sequences=True,
                     return_state=True, dropout=0.4,
                     recurrent_dropout=0.4)
(encoder_output1, state_h1, state_c1) = encoder_lstm1(enc_emb)

# Second LSTM layer of the encoder
encoder_lstm2 = LSTM(latent_dim, return_sequences=True,
                     return_state=True, dropout=0.4,
                     recurrent_dropout=0.4)
(encoder_output2, state_h2, state_c2) = encoder_lstm2(encoder_output1)

# Third LSTM layer of the encoder
encoder_lstm3 = LSTM(latent_dim, return_state=True,
                     return_sequences=True, dropout=0.4,
                     recurrent_dropout=0.4)
(encoder_outputs, state_h, state_c) = encoder_lstm3(encoder_output2)

# Setting up the decoder, using encoder_states as the initial state
decoder_inputs = Input(shape=(None, ))

# Embedding layer for decoder input sequences
dec_emb_layer = Embedding(y_voc, embedding_dim, trainable=True)
dec_emb = dec_emb_layer(decoder_inputs)

# Decoder LSTM
decoder_lstm = LSTM(latent_dim, return_sequences=True,
                    return_state=True, dropout=0.4,
                    recurrent_dropout=0.2)
(decoder_outputs, decoder_fwd_state, decoder_back_state) = \
    decoder_lstm(dec_emb, initial_state=[state_h, state_c])

# Dense layer to output probabilities over the target vocabulary
decoder_dense = TimeDistributed(Dense(y_voc, activation='softmax'))
decoder_outputs = decoder_dense(decoder_outputs)

# Defining the model
model = Model([encoder_inputs, decoder_inputs], decoder_outputs)
model.summary()

# Compiling the model with RMSprop optimizer and sparse categorical cross-entropy loss function
model.compile(optimizer='rmsprop', loss='sparse_categorical_crossentropy')

# Early stopping callback to prevent overfitting by monitoring validation loss
# If the validation loss stops decreasing for 'patience' number of epochs, training will stop
es = EarlyStopping(monitor='val_loss', mode='min', verbose=1, patience=2)

```

✓ Training Model

```

history=model.fit([x_tr, y_tr[:, :-1]],
                  y_tr.reshape(y_tr.shape[0], y_tr.shape[1], 1)[:,:1],
                  epochs=5,
                  callbacks=[es],
                  batch_size=32,
                  validation_data=([x_val, y_val[:, :-1]], y_val.reshape(y_val.shape[0], y_val.shape[1], 1)[:,:1]))

```

```

Epoch 1/5
21/21 [=====] - 301s 14s/step - loss: 4.4956 - val_loss: 2.9031
Epoch 2/5
21/21 [=====] - 288s 14s/step - loss: 3.0740 - val_loss: 2.8688
Epoch 3/5
21/21 [=====] - 288s 14s/step - loss: 2.9755 - val_loss: 2.7069
Epoch 4/5
21/21 [=====] - 288s 14s/step - loss: 2.8941 - val_loss: 2.7049

```

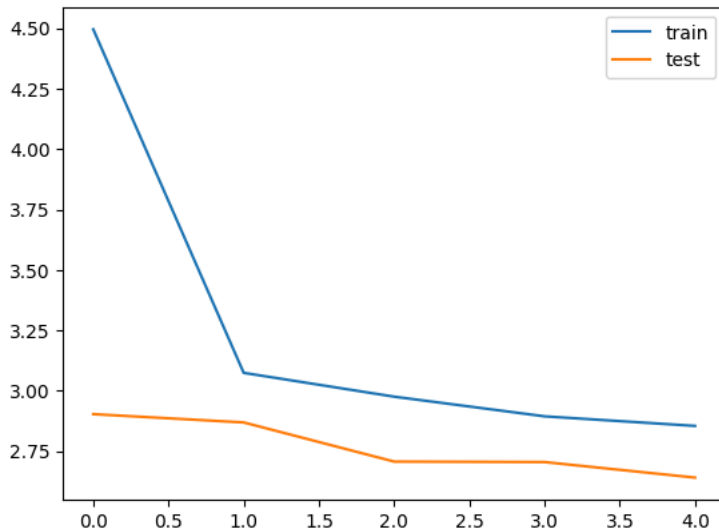
Epoch 5/5

21/21 [=====] - 291s 14s/step - loss: 2.8545 - val_loss: 2.6406

```

from matplotlib import pyplot
pyplot.plot(history.history['loss'], label='train')
pyplot.plot(history.history['val_loss'], label='test')
pyplot.legend()
pyplot.show()

```



```

# Saving the model
model.save_weights('/content/drive/MyDrive/Colab Notebooks/NLP/Project/Newss2sModel')

```

```

# Loading the model
model = model.load_weights('/content/drive/MyDrive/Colab Notebooks/NLP/Project/Newss2sModel')

```

✓ Generating Predictions and Rouge scores

```

# Creating a reverse mapping from indices to words for the target language
reverse_target_word_index = y_tokenizer.index_word

# Creating a reverse mapping from indices to words for the source language
reverse_source_word_index = x_tokenizer.index_word

# Creating a mapping from words to indices for the target language
target_word_index = y_tokenizer.word_index

```

```

# Encoding the input sequence to get the feature vector
encoder_model = Model(inputs=encoder_inputs, outputs=[encoder_outputs,
                                                    state_h, state_c])

# Decoder Setup
# Below tensors will hold the states of the previous time step
decoder_state_input_h = Input(shape=(latent_dim, ))
decoder_state_input_c = Input(shape=(latent_dim, ))
decoder_hidden_state_input = Input(shape=(max_text_len, latent_dim))

# Getting the embeddings of the decoder sequence
dec_emb2 = dec_emb_layer(decoder_inputs)

# To predict the next word in the sequence, setting the initial states to the states from the previous time step
(decoder_outputs2, state_h2, state_c2) = decoder_lstm(dec_emb2,
                                                    initial_state=[decoder_state_input_h, decoder_state_input_c])

# A dense softmax layer to generate prob dist. over the target vocabulary
decoder_outputs2 = decoder_dense(decoder_outputs2)

# Final decoder model
decoder_model = Model([decoder_inputs] + [decoder_hidden_state_input,
                                           decoder_state_input_h, decoder_state_input_c],
                     [decoder_outputs2] + [state_h2, state_c2])

def decode_sequence(input_seq):
    # Encoding the input sequence to obtain the encoder output, hidden state, and cell state.
    (e_out, e_h, e_c) = encoder_model.predict(input_seq)

    # Initializing the target sequence with a single 'start' token.
    target_seq = np.zeros((1, 1))
    target_seq[0, 0] = target_word_index['sostok'] # Start token index

    stop_condition = False # Flag to control the decoding process
    decoded_sentence = '' # Initialize the decoded sentence

    # Decoding the sequence until the stop condition is met
    while not stop_condition:
        # Predicting the next token based on the current target sequence and encoder states
        (output_tokens, h, c) = decoder_model.predict([target_seq] + [e_out, e_h, e_c])

        # Getting the index of the token with the highest probability
        sampled_token_index = np.argmax(output_tokens[0, -1, :])

        # Converting the token index to its corresponding word
        sampled_token = reverse_target_word_index[sampled_token_index]

        # Appending the word to the decoded sentence if it's not the end token
        if sampled_token != 'eostok':
            decoded_sentence += ' ' + sampled_token

        # Checking if the end token is reached or the maximum length is exceeded
        if sampled_token == 'eostok' or len(decoded_sentence.split()) >= max_summary_len - 1:
            stop_condition = True

        # Updating the target sequence with the predicted token index
        target_seq = np.zeros((1, 1))
        target_seq[0, 0] = sampled_token_index

        # Updating the encoder states for the next iteration
        (e_h, e_c) = (h, c)

    return decoded_sentence

```

```
def seq2summary(input_seq):
    '''
    This function converts a sequence of indices to a summary
    '''
    newString = ''
    # Iterating over each index in the input sequence
    for i in input_seq:
        # Checking if the index is not equal to 0 (padding), start token, or end token
        if i != 0 and i != target_word_index['sostok'] and i != target_word_index['eostok']:
            # Appending the corresponding word to the newString along with a space
            newString = newString + reverse_target_word_index[i] + ' '
    return newString

def seq2text(input_seq):
    '''
    This function converts a sequence of indices to a text
    '''
    newString = ''
    for i in input_seq:
        # Checking if the index is not equal to 0 (padding)
        if i != 0:
            # Appending the corresponding word to the newString along with a space
            newString = newString + reverse_source_word_index[i] + ' '
    return newString
```

```
# Initializing Rouge for calculating ROUGE scores
rouge = Rouge()

# Calculating ROUGE scores for each pair of original and predicted summaries
rouge_scores = []
for i in range(10):
    text = seq2text(x_tr[i])
    original_summary = seq2summary(y_tr[i])
    predicted_summary = decode_sequence(x_tr[i].reshape(1, max_text_len))

    rouge_scores.append(rouge.get_scores(predicted_summary, original_summary)[0])

# Organizing the results into a DataFrame
df_results = pd.DataFrame({
    'text': [seq2text(x_tr[i]) for i in range(10)],
    'original_summary': [seq2summary(y_tr[i]) for i in range(10)],
    'predicted_summary': [decode_sequence(x_tr[i].reshape(1, max_text_len)) for i in range(10)],
    'rouge1': [score['rouge-1']['f'] for score in rouge_scores], # ROUGE-1 scores
    'rouge2': [score['rouge-2']['f'] for score in rouge_scores] # ROUGE-2 scores
})
```

ROUGE (Recall-Oriented Understudy for Gisting Evaluation) is a set of metrics used to evaluate the quality of summaries generated by automatic summarization systems. It measures the overlap between the generated summary and the reference summary, focusing on recall, precision, and F1-score of n-grams (unigrams, bigrams, etc.)

```
# ROUGE scores from the DataFrame
rouge1_scores = df_results['rouge1']

# Calculate precision, recall, and F1-score
precision_rouge1 = rouge1_scores.mean()
recall_rouge1 = rouge1_scores.mean()
f1_rouge1 = 2 * (precision_rouge1 * recall_rouge1) / (precision_rouge1 + recall_rouge1)

# Print the results
print("ROUGE-1 Precision:", precision_rouge1)
print("ROUGE-1 Recall:", recall_rouge1)
print("ROUGE-1 F1-score:", f1_rouge1)

ROUGE-1 Precision: 0.029162156663601575
ROUGE-1 Recall: 0.029162156663601575
ROUGE-1 F1-score: 0.029162156663601575
```

```
# Seq2seq LSTM - rouge 1
df_results['rouge1'].mean()
```

```
0.029162156663601575
```

✓ T5 - Tranformer (Text-To-Text Transfer Transformer)

```
!pip install accelerate -U
```

```
Collecting accelerate
  Downloading accelerate-0.29.2-py3-none-any.whl (297 kB)
    297.4/297.4 kB 6.4 MB/s eta 0:00:00
Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.10/dist-packages (from accelerate) (1.25.2)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from accelerate) (24.0)
Requirement already satisfied: psutil in /usr/local/lib/python3.10/dist-packages (from accelerate) (5.9.5)
Requirement already satisfied: pyyaml in /usr/local/lib/python3.10/dist-packages (from accelerate) (6.0.1)
Requirement already satisfied: torch>=1.10.0 in /usr/local/lib/python3.10/dist-packages (from accelerate) (2.2.1+cu121)
Requirement already satisfied: huggingface-hub in /usr/local/lib/python3.10/dist-packages (from accelerate) (0.20.3)
Requirement already satisfied: safetensors>=0.3.1 in /usr/local/lib/python3.10/dist-packages (from accelerate) (0.4.2)
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-packages (from torch>=1.10.0->accelerate) (3.13.4)
Requirement already satisfied: typing-extensions>=4.8.0 in /usr/local/lib/python3.10/dist-packages (from torch>=1.10.0->accelerate) (4.5.0)
Requirement already satisfied: sympy in /usr/local/lib/python3.10/dist-packages (from torch>=1.10.0->accelerate) (1.12)
Requirement already satisfied: networkx in /usr/local/lib/python3.10/dist-packages (from torch>=1.10.0->accelerate) (3.3)
Requirement already satisfied: Jinja2 in /usr/local/lib/python3.10/dist-packages (from torch>=1.10.0->accelerate) (3.1.3)
Requirement already satisfied: fsspec in /usr/local/lib/python3.10/dist-packages (from torch>=1.10.0->accelerate) (2023.6.0)
Collecting nvidia-cuda-nvrtc-cu12==12.1.105 (from torch>=1.10.0->accelerate)
  Using cached nvidia_cuda_nvrtc_cu12-12.1.105-py3-none-manylinux1_x86_64.whl (23.7 MB)
Collecting nvidia-cuda-runtime-cu12==12.1.105 (from torch>=1.10.0->accelerate)
  Using cached nvidia_cuda_runtime_cu12-12.1.105-py3-none-manylinux1_x86_64.whl (823 kB)
Collecting nvidia-cuda-cupti-cu12==12.1.105 (from torch>=1.10.0->accelerate)
  Using cached nvidia_cuda_cupti_cu12-12.1.105-py3-none-manylinux1_x86_64.whl (14.1 MB)
Collecting nvidia-cudnn-cu12==8.9.2.26 (from torch>=1.10.0->accelerate)
  Using cached nvidia_cudnn_cu12-8.9.2.26-py3-none-manylinux1_x86_64.whl (731.7 MB)
Collecting nvidia-cublas-cu12==12.1.3.1 (from torch>=1.10.0->accelerate)
  Using cached nvidia_cublas_cu12-12.1.3.1-py3-none-manylinux1_x86_64.whl (410.6 MB)
Collecting nvidia-cufft-cu12==11.0.2.54 (from torch>=1.10.0->accelerate)
  Using cached nvidia_cufft_cu12-11.0.2.54-py3-none-manylinux1_x86_64.whl (121.6 MB)
Collecting nvidia-curand-cu12==10.3.2.106 (from torch>=1.10.0->accelerate)
  Using cached nvidia_curand_cu12-10.3.2.106-py3-none-manylinux1_x86_64.whl (56.5 MB)
Collecting nvidia-cusolver-cu12==11.4.5.107 (from torch>=1.10.0->accelerate)
  Using cached nvidia_cusolver_cu12-11.4.5.107-py3-none-manylinux1_x86_64.whl (124.2 MB)
Collecting nvidia-cuspars-cu12==12.1.0.106 (from torch>=1.10.0->accelerate)
  Using cached nvidia_cuspars-cu12-12.1.0.106-py3-none-manylinux1_x86_64.whl (196.0 MB)
Collecting nvidia-nccl-cu12==2.19.3 (from torch>=1.10.0->accelerate)
  Using cached nvidia_nccl_cu12-2.19.3-py3-none-manylinux1_x86_64.whl (166.0 MB)
Collecting nvidia-nvtx-cu12==12.1.105 (from torch>=1.10.0->accelerate)
  Using cached nvidia_nvtx_cu12-12.1.105-py3-none-manylinux1_x86_64.whl (99 kB)
Requirement already satisfied: triton==2.2.0 in /usr/local/lib/python3.10/dist-packages (from torch>=1.10.0->accelerate) (2.2.0)
Collecting nvidia-nvjitlink-cu12 (from nvidia-cusolver-cu12==11.4.5.107->torch>=1.10.0->accelerate)
  Using cached nvidia_nvjitlink_cu12-12.4.127-py3-none-manylinux2014_x86_64.whl (21.1 MB)
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (from huggingface-hub->accelerate) (2.31.0)
Requirement already satisfied: tqdm>=4.42.1 in /usr/local/lib/python3.10/dist-packages (from huggingface-hub->accelerate) (4.66.1)
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.10/dist-packages (from Jinja2->torch>=1.10.0->accelerate) (2.1.2)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests->huggingface-hub->accelerate) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests->huggingface-hub->accelerate) (3.6)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests->huggingface-hub->accelerate) (2.0.7)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests->huggingface-hub->accelerate) (2023.7.22)
Requirement already satisfied: mpmath>=0.19 in /usr/local/lib/python3.10/dist-packages (from sympy->torch>=1.10.0->accelerate) (3.1.0)
Installing collected packages: nvidia-nvtx-cu12, nvidia-nvjitlink-cu12, nvidia-nccl-cu12, nvidia-curand-cu12, nvidia-cufft-cu12, nvidia-cuspars-cu12, nvidia-cusolver-cu12, nvidia-cuda-cupti-cu12, nvidia-cuda-runtime-cu12, nvidia-cuda-nvrtc-cu12, accelerate
Successfully installed accelerate-0.29.2 nvidia-cublas-cu12-12.1.3.1 nvidia-cuda-cupti-cu12-12.1.105 nvidia-cuda-nvrtc-cu12-
```

```
from transformers import AutoTokenizer
from transformers import DataCollatorForSeq2Seq, AutoModelForSeq2SeqLM, Seq2SeqTrainingArguments, Seq2SeqTrainer
import accelerate
from transformers import Seq2SeqTrainer
from transformers import TFAutoModelForSeq2SeqLM
from rouge import Rouge
```

```
# Getting the data
from datasets import load_dataset
multi_news = load_dataset("multi_news", split="test")
```

```
# # Splitting the data into train and test for T5
# from sklearn.model_selection import train_test_split

multi_news = multi_news.train_test_split(test_size=0.2)
```

✓ Preparing Tokenizing

```
# Load the tokenizer for the T5-small model from the Hugging Face model hub
tokenizer = AutoTokenizer.from_pretrained("t5-small")
```

```
prefix = "summarize: "
```

```
def preprocess_function(examples):
    # Adding the prefix to each document for summarization
    inputs = [prefix + doc for doc in examples["document"]]

    # Tokenizing the inputs using the T5 tokenizer
    # Setting max_length and truncation parameters for both inputs and labels
    model_inputs = tokenizer(inputs, max_length=1024, truncation=True)

    # Tokenizing the summaries using the T5 tokenizer and extract the input_ids
    labels = tokenizer(text=examples["summary"], max_length=128, truncation=True)

    # Assigning the tokenized summaries (input_ids) to the "labels" key in the model_inputs dictionary
    model_inputs["labels"] = labels["input_ids"]

    return model_inputs
```

```
tokenized_multi_news = multi_news.map(preprocess_function, batched=True)
# tokenized_multi_news = preprocess_function(train_data)
```

Map: 100% 4497/4497 [00:12<00:00, 362.45 examples/s]

Map: 100% 1125/1125 [00:03<00:00, 347.56 examples/s]

```
tokenized_multi_news['train'].to_pandas()
```

	document	summary	input_ids	attention_mask	labels
0	Kris Humphries to File for Separation – Not Di...	– Poor Kim: Not only have her fairy tale dream...	[21603, 10, 9375, 3455, 7656, 2593, 12, 7344, ...	[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...	[3, 104, 21309, 6777, 10, 933, 163, 43, 160, 1...
1	A record 400 farmers attacked in 2017, between...	– The South African government has begun the c...	[21603, 10, 71, 1368, 4837, 7208, 17263, 16, 4...	[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...	[3, 104, 37, 1013, 3850, 789, 65, 11173, 8, 15...
2	BERKLEY (WXYZ) - October 6, 1966. \n \n \n \n ...	– A Detroit man who long pondered the fate of ...	[21603, 10, 3, 12920, 439, 3765, 476, 41, 518,...	[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...	[3, 104, 71, 11901, 388, 113, 307, 3, 31462, 2...
3	Scott Weiland Will Filed ... Ex-Wife Wants to ...	– Scott Weiland's death could end up just as f...	[21603, 10, 4972, 101, 173, 232, 2003, 7344, 2...	[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...	[3, 104, 4972, 101, 173, 232, 31, 7, 1687, 228...
4	His grandfather had lovingly given him his nam...	– Saddam Hussain's name is making it difficult...	[21603, 10, 978, 18573, 141, 6330, 120, 787, 3...	[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...	[3, 104, 18875, 7812, 13674, 7, 9, 77, 31, 7, ...

```
import accelerate
```

```
# Loading Model
# Data collator for sequence-to-sequence (seq2seq) tasks, specifically designed for T5 model
data_collator = DataCollatorForSeq2Seq(tokenizer=tokenizer, model='t5-small')
```

```
# Loading the pre-trained T5-small model for sequence-to-sequence language modeling
# from the Hugging Face model hub
model = AutoModelForSeq2SeqLM.from_pretrained("t5-small")
```

```
# Hyperparameters for T5
training_args = Seq2SeqTrainingArguments(
    output_dir="./results",
    evaluation_strategy="epoch",
    learning_rate=2e-5,
    per_device_train_batch_size=10,
    per_device_eval_batch_size=10,
    weight_decay=0.01,
    save_total_limit=3,
    num_train_epochs=10,
    fp16=True,
)
```

Initializing Trainer

```
# Initializing the Seq2SeqTrainer for training and evaluation
```

```
trainer = Seq2SeqTrainer(
    model=model,
    args=training_args,
    train_dataset=tokenized_multi_news["train"],
    eval_dataset=tokenized_multi_news["test"],
    tokenizer=tokenizer,
    data_collator=data_collator,
)
```

```
# Training the T5 Model
trainer.train()
```

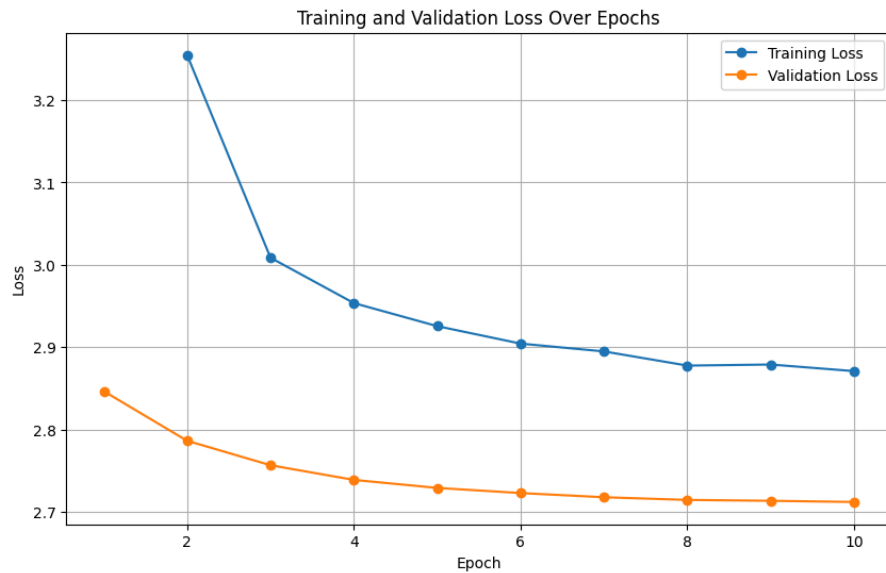
 [4500/4500 16:13, Epoch 10/10]

Epoch	Training Loss	Validation Loss
1	No log	2.846481
2	3.254700	2.786284
3	3.008600	2.756580
4	2.953500	2.738764
5	2.925500	2.729048
6	2.904300	2.722795
7	2.894800	2.717708
8	2.877600	2.714469
9	2.878900	2.713377
10	2.870900	2.711978

```
TrainOutput(global_step=4500, training_loss=2.952074951171875, metrics=
{'train_runtime': 975.317, 'train_samples_per_second': 46.108,
 'train_steps_per_second': 4.614, 'total_flos': 1.217264162439168e+16})
```

```
# t5_loss = {
#     'Epoch': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10],
#     'Training Loss': [np.nan, 3.2547, 3.0086, 2.9535, 2.9255, 2.9043, 2.8948, 2.8776, 2.8789, 2.8709],
#     'Validation Loss': [2.846481, 2.786284, 2.75658, 2.738764, 2.729048, 2.722795, 2.717708, 2.714469, 2.713377, 2.711978]
# }

# t5_loss_df = pd.DataFrame(t5_loss)
plt.figure(figsize=(10, 6))
plt.plot(t5_loss_df['Epoch'], t5_loss_df['Training Loss'], marker='o', label='Training Loss')
plt.plot(t5_loss_df['Epoch'], t5_loss_df['Validation Loss'], marker='o', label='Validation Loss')
plt.title('Training and Validation Loss Over Epochs')
plt.xlabel('Epoch')
plt.ylabel('Loss')
plt.legend()
plt.grid(True)
plt.show()
```

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

Mounted at /content/drive

```
# # Saving the model in drive
# model_dir = "/content/drive/MyDrive/Colab Notebooks/NLP/Project/TModel"
# trainer.save_model(model_dir)
```

```
# print("Model saved successfully to Google Drive.")
```

Model saved successfully to Google Drive.

```
# Loading the model from the drive
model_dir = "/content/drive/MyDrive/Colab Notebooks/NLP/Project/TModel"
loaded_model = TFAutoModelForSeq2SeqLM.from_pretrained(model_dir)
```

```
print("Model loaded successfully from Google Drive.")
```

All PyTorch model weights were used when initializing TFT5ForConditionalGeneration.

All the weights of TFT5ForConditionalGeneration were initialized from the PyTorch model.

If your task is similar to the task the model of the checkpoint was trained on, you can already use TFT5ForConditionalGeneration. Model loaded successfully from Google Drive.

✓ Generating predictions and rouge scores

```
def predict_summary(document):
    device = model.device
    # Tokenizing the input document using the tokenizer
    tokenized = tokenizer([document], truncation=True, padding='longest', return_tensors='pt')
    tokenized = {k: v.to(device) for k, v in tokenized.items()}
    tokenized_result = model.generate(**tokenized, max_length=128) # Generating summary
    tokenized_result = tokenized_result.to('cpu')
    predicted_summary = tokenizer.decode(tokenized_result[0]) # convert the generated tokens into Text

    return predicted_summary
```

```
def get_rouge_scores(actual_summary, predicted_summary):
    '''This function generates the rouge scores'''
    rouge = Rouge()
    scores = rouge.get_scores(predicted_summary, actual_summary)
    return [scores[0]['rouge-1']['f'], scores[0]['rouge-2']['f'], scores[0]['rouge-l']['f']]
```

```
dataset = load_dataset("multi_news")
val_data = dataset['validation'].to_pandas()
```

```
val_data.head()
```

	document	summary
0	Whether a sign of a good read; or a comment on...	– The Da Vinci Code has sold so many copies –th...
1	The deaths of three American soldiers in Afgha...	– A major snafu has hit benefit payments to st...
2	DUBAI Al Qaeda in Yemen has claimed responsibi...	– Yemen-based al-Qaeda in the Arabian Peninsul...

Cambridge Analytica, a data firm that worked

Next steps:

[Generate code with val_data](#)

[View recommended plots](#)

```
from rouge import Rouge

# Initialize Rouge
rouge = Rouge()

# Function to calculate ROUGE scores
def get_rouge_scores(human_summary, pred_summary):
    scores = rouge.get_scores(human_summary, pred_summary)
    rouge1_precision = scores[0]["rouge-1"]["p"]
    rouge1_recall = scores[0]["rouge-1"]["r"]
    rouge1_f1score = scores[0]["rouge-1"]["f"]
    return rouge1_precision, rouge1_recall, rouge1_f1score

# Calculate ROUGE scores for each summary in the validation data
precision_list = []
recall_list = []
f1score_list = []

for i in range(len(val_data)):
    human_summary = val_data.loc[i]['summary']
    pred_summary = val_data.loc[i]['pred_summary']
    precision, recall, f1score = get_rouge_scores(human_summary, pred_summary)
    precision_list.append(precision)
    recall_list.append(recall)
    f1score_list.append(f1score)

# Create a DataFrame to store the ROUGE scores
rouge_scores_df = pd.DataFrame({
    'Precision': precision_list,
    'Recall': recall_list,
    'F1-score': f1score_list
})

# Display the DataFrame
print(rouge_scores_df)
```

	Precision	Recall	F1-score
0	0.133758	0.466667	0.207921
1	0.112994	0.434783	0.179372
2	0.104000	0.361111	0.161491
3	0.106796	0.407407	0.169231
4	0.090909	0.311111	0.140704
5	0.088372	0.441860	0.147287
6	0.131783	0.326923	0.187845
7	0.083871	0.361111	0.136126
8	0.117647	0.545455	0.193548
9	0.156977	0.642857	0.252336
10	0.063584	0.550000	0.113990
11	0.145833	0.466667	0.222222
12	0.088398	0.533333	0.151659
13	0.063380	0.200000	0.096257

```

14 0.019355 0.093750 0.032086
15 0.074074 0.421053 0.125984
16 0.026201 0.181818 0.045802
17 0.152941 0.433333 0.226087
18 0.150000 0.500000 0.230769
19 0.087838 0.406250 0.144444
20 0.226027 0.673469 0.338462
21 0.141304 0.530612 0.223176
22 0.046053 0.175000 0.072917
23 0.114943 0.465116 0.184332
24 0.040462 0.411765 0.073684
25 0.067633 0.341463 0.112903
26 0.076190 0.228571 0.114286
27 0.092025 0.348837 0.145631
28 0.232558 0.750000 0.355030
29 0.079710 0.550000 0.139241
30 0.066225 0.238095 0.103627
31 0.090909 0.406250 0.148571
32 0.157233 0.555556 0.245098
33 0.129496 0.580645 0.211765
34 0.149351 0.522727 0.232323
35 0.094488 0.375000 0.150943
36 0.121622 0.562500 0.200000
37 0.086705 0.576923 0.150754
38 0.099476 0.413043 0.160338
39 0.073770 0.321429 0.120000
40 0.144231 0.306122 0.196078
41 0.067568 0.250000 0.106383
42 0.129032 0.533333 0.207792
43 0.125874 0.600000 0.208092
44 0.130435 0.405405 0.197368
45 0.099338 0.405405 0.159574
46 0.060000 0.225000 0.094737
47 0.096154 0.357143 0.151515
48 0.143836 0.420000 0.214286
49 0.169492 0.487805 0.251572

```

```
rouge_scores_df.mean()
```

```

Precision    0.107017
Recall       0.422054
F1-score     0.168713
dtype: float64

```

```

# rouge score of validation data
val_data = val_data.iloc[:50,:]
from tqdm import tqdm

rouge1_scores = []
pred_summary_list = []

for i in tqdm(range(50)):

    doc = val_data.loc[i]['document']
    pred_summary = predict_summary(doc)
    human_summary = val_data.loc[i]['summary']

    score = get_rouge_scores(human_summary, pred_summary)

    rouge1_scores.append(score[0])
    pred_summary_list.append(pred_summary)




val_data["pred_summary"] = pred_summary_list

val_data['rouge1'] = rouge1_scores

val_data

```

100%|██████████| 50/50 [01:04<00:00, 1.28s/it]

	document	summary	pred_summary	rouge1	
0	Whether a sign of a good read; or a comment on...	– The Da Vinci Code has sold so many copies—th...	<pad><extra_id_0>, <extra_id_1>, <extra_id_2>,<e...	0.207921	 
1	The deaths of three American soldiers in Afgha...	– A major snafu has hit benefit payments to st...	<pad><extra_id_0> is a national disgrace, one ...	0.179372	
2	DUBAI Al Qaeda in Yemen has claimed responsibi...	– Yemen-based al-Qaeda in the Arabian Peninsul...	<pad><extra_id_0> has claimed responsibility f...	0.161491	
3	Cambridge Analytica, a data firm that worked f...	– Cambridge Analytica is calling it quits. The...	<pad><extra_id_0>, <extra_id_1>, <extra_id_2>, a...	0.169231	
4	The N.S.A.'s Evolution: The National Security ...	– A lengthy report in the New York Times, base...	<pad><extra_id_0>, <extra_id_1>, <extra_id_2>,<e...	0.140704	