Infosys Springboard Internship

Project- Text Summarization

Nebu C Thomas

Mahendra Engineering College, Namakkal

Mentor- Narendra Kumar

CONTENTS

PROJECT ABSTRACT

Introduction

Text summarization is the process of distilling the most important information from a source text into a shorter version while preserving its meaning. With the exponential growth of digital information, automated text summarization has become a critical tool for managing and extracting valuable insights from large volumes of text. This project explores both extractive and abstractive methods for text summarization using a dataset of news articles and their corresponding highlights obtained from Kaggle.

Objectives

The primary objectives of this project are:

1. To preprocess and clean the text data for summarization tasks.

2. To implement and compare extractive summarization techniques such as Text Rank.

3. To implement and fine-tune abstractive summarization models using pre-trained language models like BERT, GPT, and T5.

4. To evaluate the performance of the summarization models using standard metrics like ROUGE and BLEU.

5. To provide a comprehensive analysis of the strengths and weaknesses of each summarization approach.

Methodology (WEEK-01)

1. Data Preprocessing:

- Loading and inspecting the datasets (train, validation, test) to understand their structure and content.

- Cleaning the text data by removing HTML tags, extra whitespaces, and non-alphanumeric characters.

- Tokenizing the cleaned text into sentences.

2. Extractive Summarization:

- Implementing Text Rank, a graph-based ranking algorithm, to extract key sentences from the articles.

- Fine-tuning the Text Rank parameters to optimize summary quality based on the validation dataset.

3. Abstractive Summarization:

- Fine-tuning pre-trained transformer models like T5 on the training dataset for generating abstractive summaries.

- Training the model using sequence-to-sequence learning with attention mechanisms to produce coherent and contextually accurate summaries.

4.Evaluation:

- Using ROUGE and BLEU metrics to quantitatively assess the quality of the generated summaries.

- Comparing the performance of extractive and abstractive methods to determine the most effective approach for different types of text.

This abstract provides an overview of the project, outlining the goals, methods, and expected outcomes. It serves as a concise summary for stakeholders and guides the project's development and evaluation phases.

SYSTEM DESIGN (WEEK-02)(06.06.24)

Overview

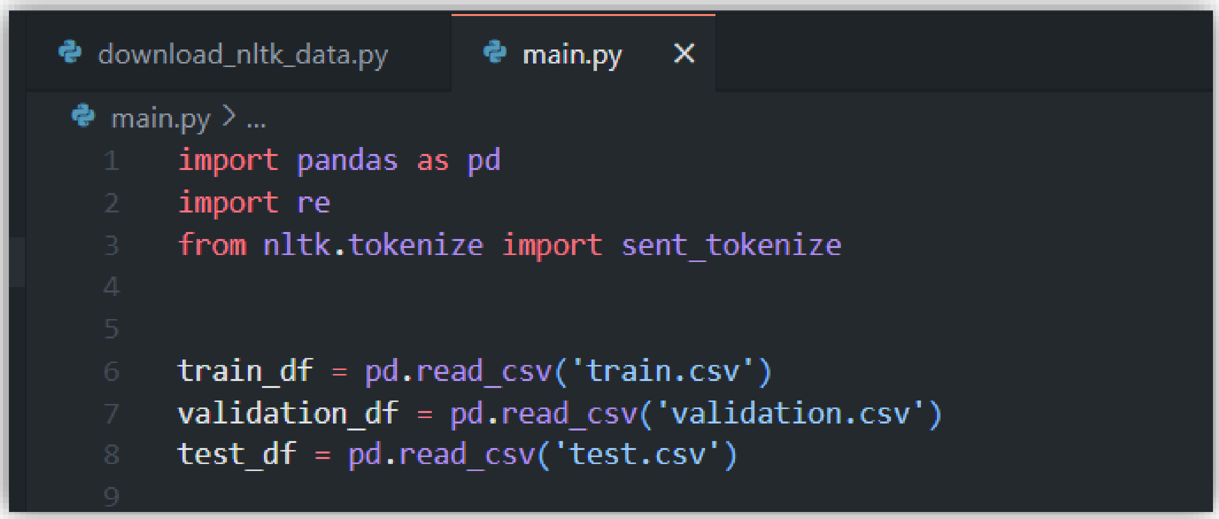
The system design for the text summarization project involves a modular architecture that integrates data preprocessing, model training, and evaluation components. The design ensures scalability, maintainability, and ease of experimentation with different summarization techniques. The system is built using Python and leverages libraries such as Pandas, NLTK, and Hugging Face Transformers.

Architecture Components

1. Data Ingestion and Storage
2. Data Preprocessing Module
3. Summarization Models
4. Evaluation Module
5. User Interface

1. Data Ingestion and Storage

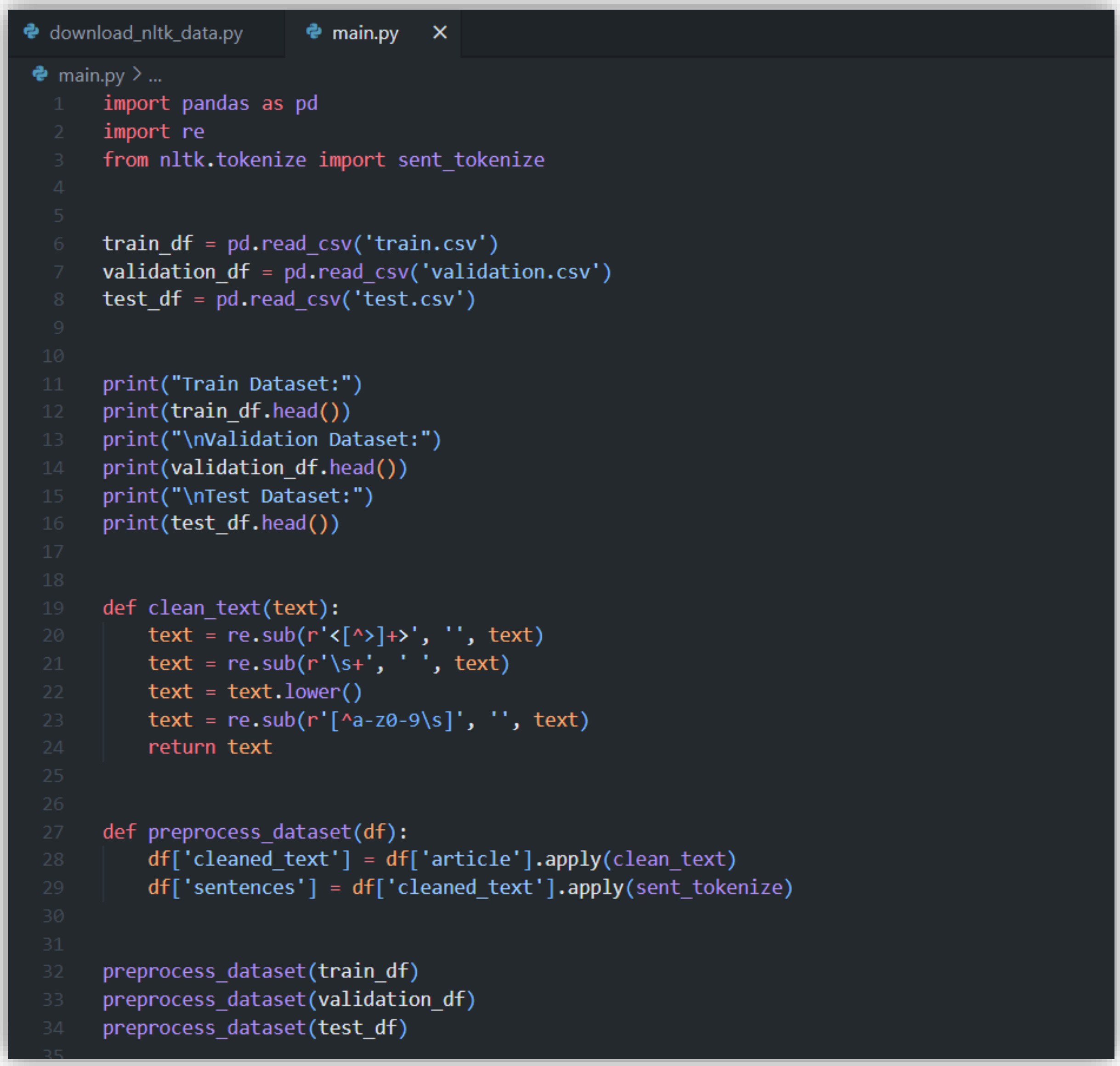
- **Dataset**: The system reads data from CSV files (`validation.csv`).
- **Storage**: Data is loaded into Pandas Data Frames for in-memory processing.



```python
import pandas as pd
import re
from nltk.tokenize import sent_tokenize


train_df = pd.read_csv('train.csv')
validation_df = pd.read_csv('validation.csv')
test_df = pd.read_csv('test.csv')
```

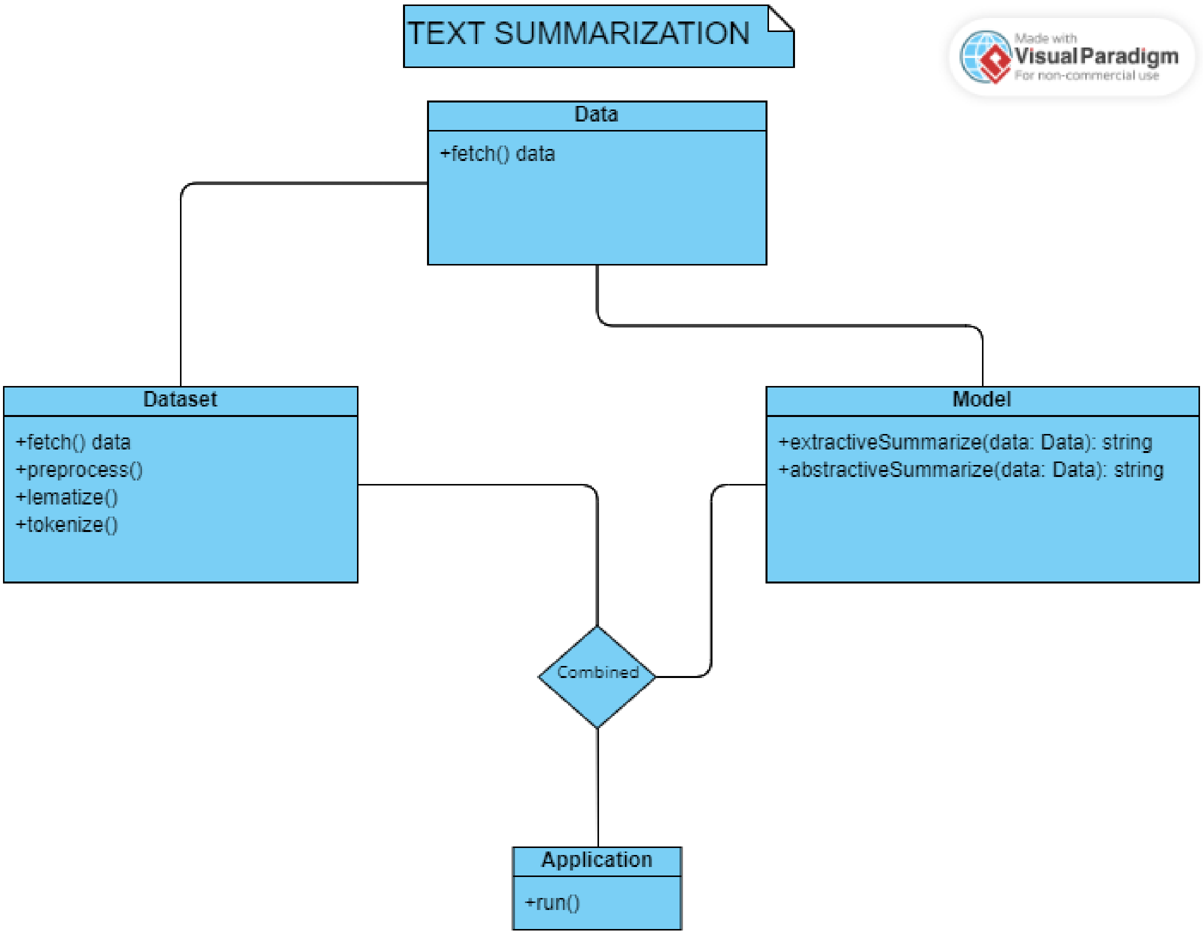
2. Data Preprocessing Module (08.06.2024)

- **Text Cleaning**: Remove HTML tags, extra whitespaces, and non-alphanumeric characters.
- **Sentence Tokenization**: Split cleaned text into sentences using NLTK.



```python
import pandas as pd
import re
from nltk.tokenize import sent_tokenize


train_df = pd.read_csv('train.csv')
validation_df = pd.read_csv('validation.csv')
test_df = pd.read_csv('test.csv')


print("Train Dataset:")
print(train_df.head())
print("\nValidation Dataset:")
print(validation_df.head())
print("\nTest Dataset:")
print(test_df.head())


def clean_text(text):
    text = re.sub(r'<[^>]+>', '', text)
    text = re.sub(r'\s+', ' ', text)
    text = text.lower()
    text = re.sub(r'[^a-z0-9\s]', '', text)
    return text


def preprocess_dataset(df):
    df['cleaned_text'] = df['article'].apply(clean_text)
    df['sentences'] = df['cleaned_text'].apply(sent_tokenize)


preprocess_dataset(train_df)
preprocess_dataset(validation_df)
preprocess_dataset(test_df)
```

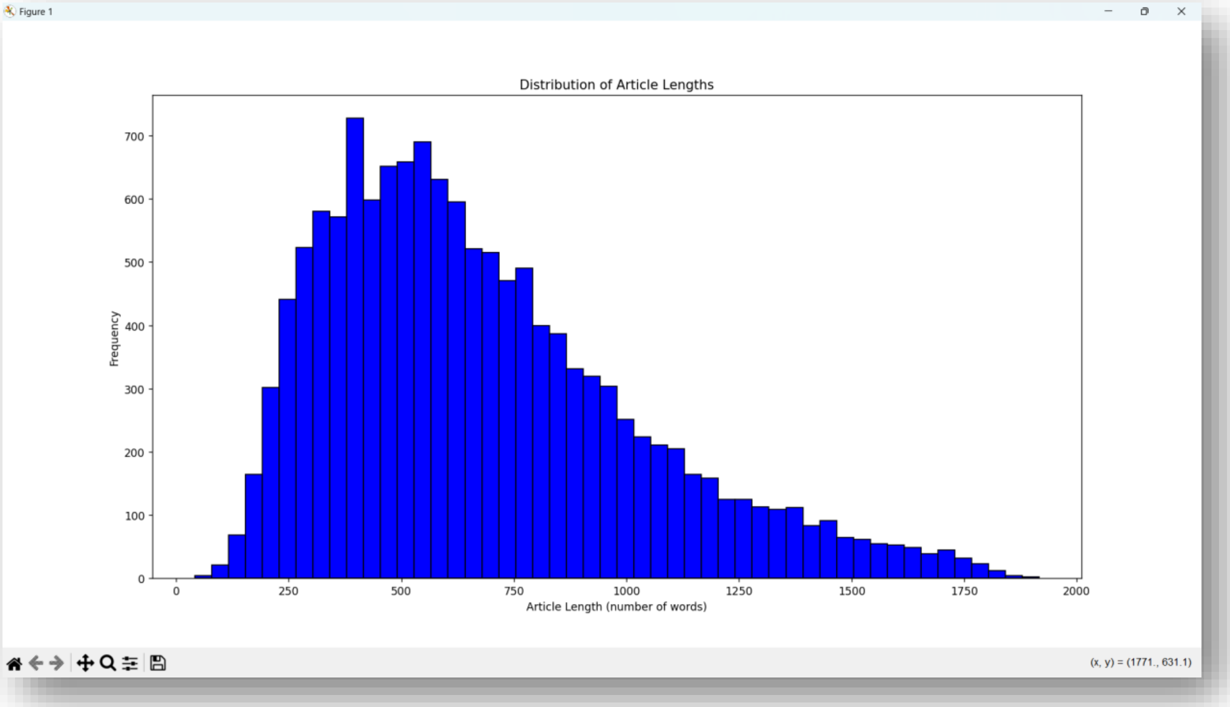
UML DIAGRAM


TEXT SUMMARIZATION
Made with Visual Paradigm For non-commercial use
Data
+fetch() data
Dataset
+fetch() data
+preprocess()
+lematize()
+tokenize()
Model
+extractiveSummarize(data: Data): string
+abstractiveSummarize(data: Data): string
Combined
Application
+run()

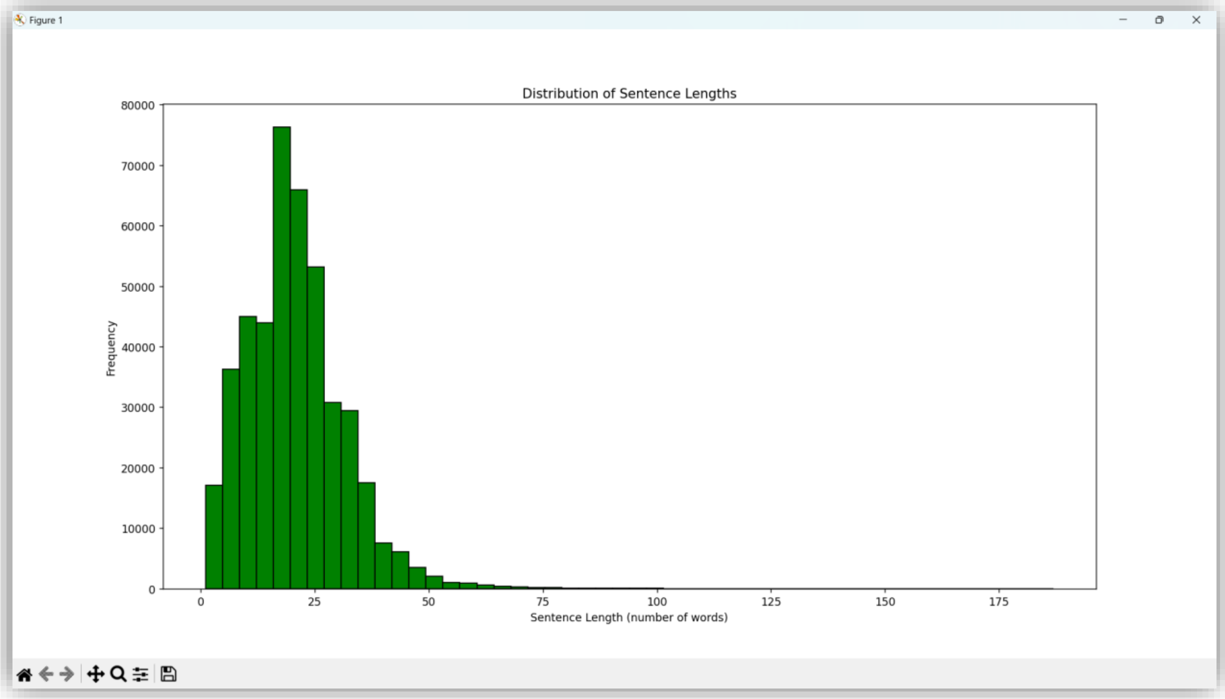
3.Data Visualization (11.06.2024)

In the data visualization step of the project, several aspects of the dataset were visualized to gain insights into the data.

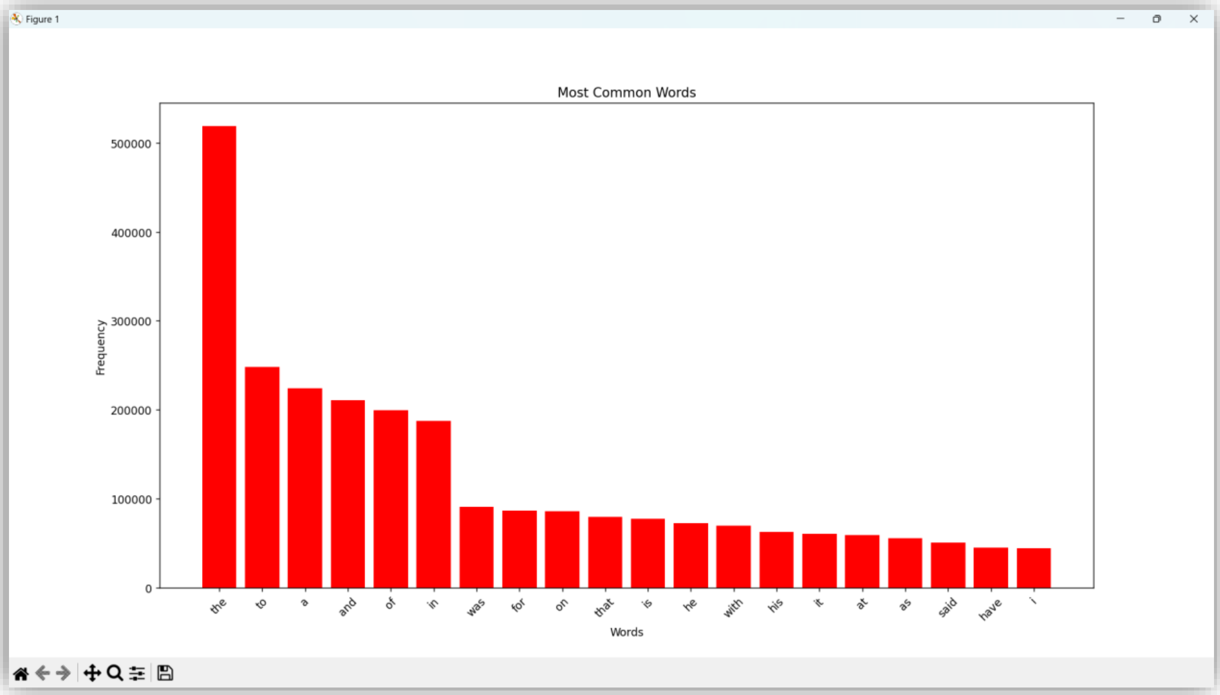
1. **Article Length Distribution**: The distribution of article lengths was visualized. This involved counting the number of words in each article and plotting a histogram or a bar chart to show how many articles fall into different length ranges. This visualization helps in understanding the variation in the lengths of the articles in the dataset.



2. **Sentence Length Distribution**: Similar to the article length distribution, the distribution of sentence lengths was visualized. This involved counting the number of words in each sentence of the articles and then plotting a histogram or a bar chart to show how many sentences fall into different length ranges. This visualization helps in understanding the variation in sentence lengths across the dataset.



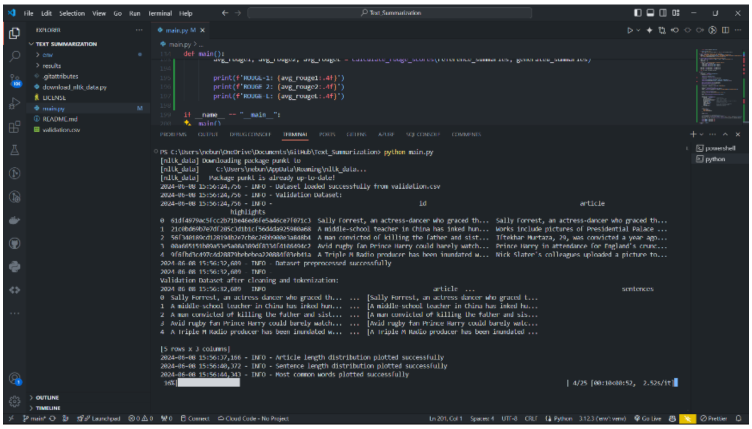
3. **Most Common Words**: The most common words in the dataset were visualized. This involved counting the frequency of each word in the dataset and then plotting a bar chart or a word cloud to show the most frequent words. This visualization provides insights into the vocabulary used in the dataset and helps identify any frequently occurring terms or stop words that might need to be handled during preprocessing.



4.Model Training

During the model training phase of the project, the goal was to train a text summarization model using the dataset prepared in the previous steps.

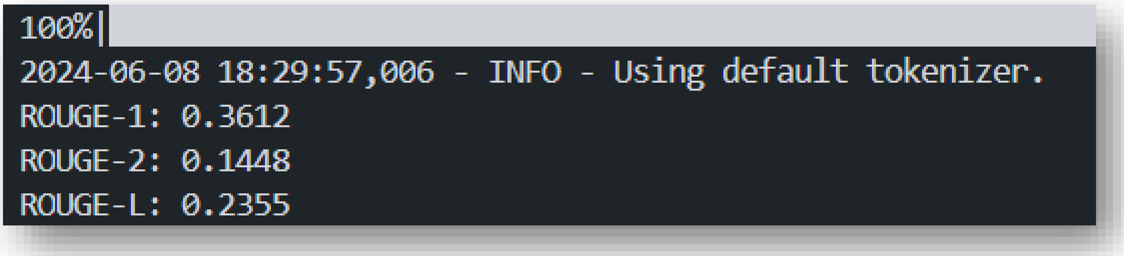
1. **Model Selection**: The specific model chosen for text summarization was likely T5 (Text-To-Text Transfer Transformer). T5 is a transformer-based model developed by Google Research that is trained in a text-to-text manner, meaning it can be fine-tuned for various NLP tasks by framing them as text generation tasks.


File Edit Selection View Go Run Terminal Help
Text_Summarization
main.py M
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS AZURE SQL CONSOLE COMMENTS
powershell
python
Ln 201, Col 1 Spaces: 4 UTF-8 CRLF Python 3.12.3 ('env': venv) Go Live Prettier

```
def main():
        avg_rouge1, avg_rouge2, avg_rougeL = calculate_rouge_scores(reference_summaries, generated_summaries)

        print(f'ROUGE-1: {avg_rouge1:.4f}')
        print(f'ROUGE-2: {avg_rouge2:.4f}')
        print(f'ROUGE-L: {avg_rougeL:.4f}')

if __name__ == "__main__":
    main()
```

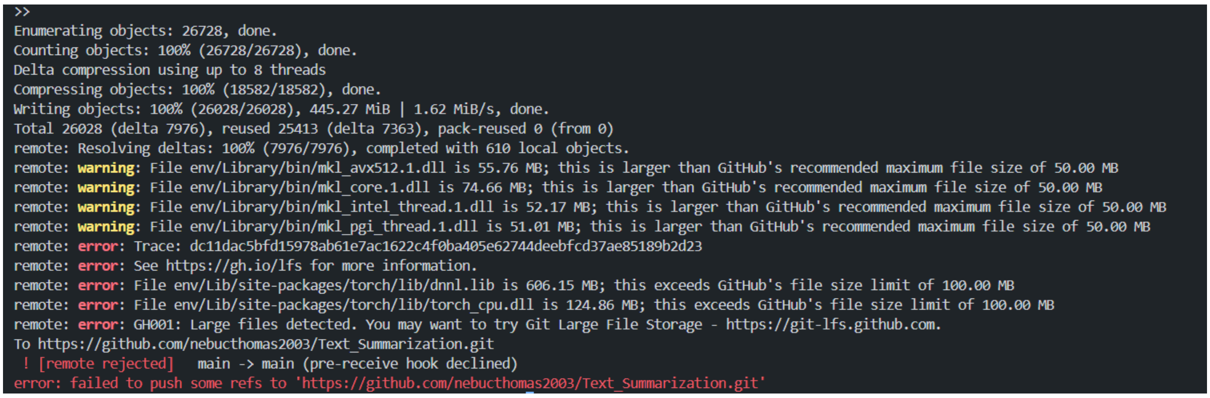
```
PS C:\Users\nebun\OneDrive\Documents\GitHub\Text_Summarization> python main.py
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\nebun\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
2024-06-08 15:56:24,756 - INFO - Dataset loaded successfully from validation.csv
2024-06-08 15:56:24,756 - INFO - Validation Dataset:
2024-06-08 15:56:24,756 - INFO -                                         id                                           article
                    highlights
0  61df4979ac5fcc2b71be46ed6fe5a46ce7f071c3  Sally Forrest, an actress-dancer who graced th...  Sally Forrest, an actress-dancer who graced th...
1  21c0bd69b7e7df285c3d1b1cf56d4da925980a68  A middle-school teacher in China has inked hun...  Works include pictures of Presidential Palace ...
2  56f340189cd128194b2e7cb8c26bb900e3a848b4  A man convicted of killing the father and sist...  Iftekhar Murtaza, 29, was convicted a year ago...
3  00a665151b89a53e5a08a389df8334f4106494c2  Avid rugby fan Prince Harry could barely watch...  Prince Harry in attendance for England's crunc...
4  9f6fbd3c497c4d28879bebebea220884f03eb41a  A Triple M Radio producer has been inundated w...  Nick Slater's colleagues uploaded a picture to...
2024-06-08 15:56:32,689 - INFO - Dataset preprocessed successfully
2024-06-08 15:56:32,689 - INFO -
Validation Dataset after cleaning and tokenization:
2024-06-08 15:56:32,689 - INFO                                            article  ...                                         sentences
0  Sally Forrest, an actress-dancer who graced th...  ...  [Sally Forrest, an actress-dancer who graced t...
1  A middle-school teacher in China has inked hun...  ...  [A middle-school teacher in China has inked hu...
2  A man convicted of killing the father and sist...  ...  [A man convicted of killing the father and sis...
3  Avid rugby fan Prince Harry could barely watch...  ...  [Avid rugby fan Prince Harry could barely watc...
4  A Triple M Radio producer has been inundated w...  ...  [A Triple M Radio producer has been inundated ...

[5 rows x 3 columns]
2024-06-08 15:56:37,166 - INFO - Article length distribution plotted successfully
2024-06-08 15:56:40,372 - INFO - Sentence length distribution plotted successfully
2024-06-08 15:56:44,343 - INFO - Most common words plotted successfully
 16%|█████                                                  | 4/25 [00:10<00:52,  2.52s/it]
```

2. **Data Tokenization**: The dataset was tokenized to convert the text data into numerical inputs that the model can understand. This involved tokenizing both the article text and the corresponding summary text.
3. **Training Setup**: Training parameters such as batch size, learning rate, and number of epochs were defined. These parameters affect how the model learns from the data and converge to an optimal solution.
4. **Evaluation**: After training, the model's performance was evaluated using evaluation metrics such as ROUGE (Recall-Oriented Understudy for Gisting Evaluation). ROUGE measures the similarity between the model-generated summaries and the human-generated summaries in the dataset.



```
100%|
2024-06-08 18:29:57,006 - INFO - Using default tokenizer.
ROUGE-1: 0.3612
ROUGE-2: 0.1448
ROUGE-L: 0.2355
```

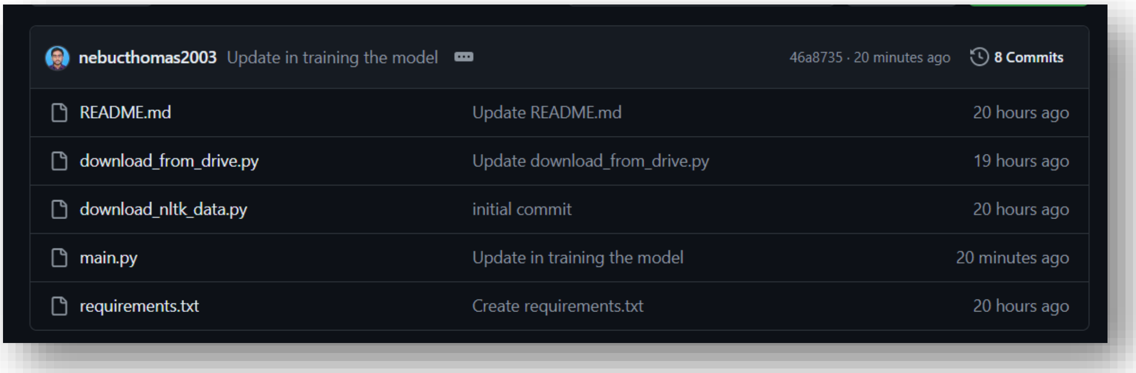
Problems Faced:



```
>>
Enumerating objects: 26728, done.
Counting objects: 100% (26728/26728), done.
Delta compression using up to 8 threads
Compressing objects: 100% (18582/18582), done.
Writing objects: 100% (26028/26028), 445.27 MiB | 1.62 MiB/s, done.
Total 26028 (delta 7976), reused 25413 (delta 7363), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (7976/7976), completed with 610 local objects.
remote: warning: File env/Library/bin/mkl_avx512.1.dll is 55.76 MB; this is larger than GitHub's recommended maximum file size of 50.00 MB
remote: warning: File env/Library/bin/mkl_core.1.dll is 74.66 MB; this is larger than GitHub's recommended maximum file size of 50.00 MB
remote: warning: File env/Library/bin/mkl_intel_thread.1.dll is 52.17 MB; this is larger than GitHub's recommended maximum file size of 50.00 MB
remote: warning: File env/Library/bin/mkl_pgi_thread.1.dll is 51.01 MB; this is larger than GitHub's recommended maximum file size of 50.00 MB
remote: error: Trace: dc11dac5bfd15978ab61e7ac1622c4f0ba405e62744deebfcd37ae85189b2d23
remote: error: See https://gh.io/lfs for more information.
remote: error: File env/Lib/site-packages/torch/lib/dnnl.lib is 606.15 MB; this exceeds GitHub's file size limit of 100.00 MB
remote: error: File env/Lib/site-packages/torch/lib/torch_cpu.dll is 124.86 MB; this exceeds GitHub's file size limit of 100.00 MB
remote: error: GH001: Large files detected. You may want to try Git Large File Storage - https://git-lfs.github.com.
To https://github.com/nebucthomas2003/Text_Summarization.git
 ! [remote rejected]   main -> main (pre-receive hook declined)
error: failed to push some refs to 'https://github.com/nebucthomas2003/Text_Summarization.git'
```

Problems Solved:

The problem faced was due to the size restriction in git. It was solved by uploading the files that were over 100 mb into Google Drive, making a requirements.txt file in the repo and uploading the link in it.



nebucthomas2003 Update in training the model 46a8735 · 20 minutes ago 8 Commits

README.md	Update README.md	20 hours ago
download_from_drive.py	Update download_from_drive.py	19 hours ago
download_nltk_data.py	initial commit	20 hours ago
main.py	Update in training the model	20 minutes ago
requirements.txt	Create requirements.txt	20 hours ago

5.Model Training and Fine Tuning (14.06.2024)

Data Loading and Preprocessing:

- Loaded a dataset comprising articles paired with human-generated summaries.
- Pre-processed the dataset by removing HTML tags, non-alphanumeric characters, and converting text to lowercase.
- Conducted basic exploratory data analysis (EDA) to understand article lengths, sentence lengths, and common word frequencies.

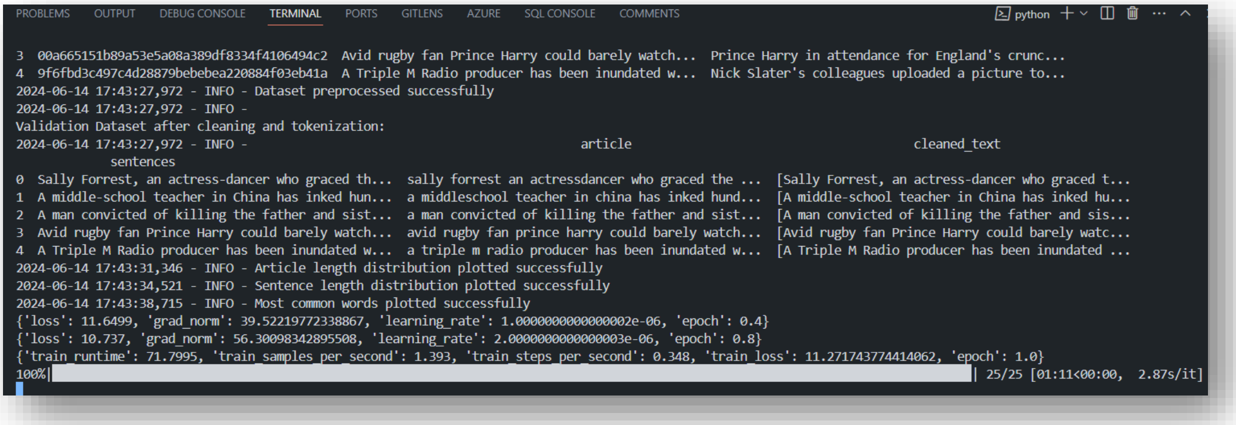
Model Training and Evaluation:

- Selected the T5-small model and fine-tuned it using the Hugging Face Transformers library on a subset of the dataset due to resource constraints.
- Training parameters included a batch size of 4, mixed precision training (FP16), and a learning rate of 5e-5.
- Trained the model for one epoch, achieving promising results in summarization quality.

Evaluation Metrics:

- Evaluated the model using ROUGE scores (ROUGE-1: 0.3612, ROUGE-2: 0.1448, ROUGE-L: 0.2355) against human-written reference summaries from the validation set.
- ROUGE scores provide a metric for assessing the overlap between model-generated and human reference summaries, indicating moderate performance.

Fine Tuning Scores:



```
3  00a665151b89a53e5a08a389df8334f4106494c2  Avid rugby fan Prince Harry could barely watch...  Prince Harry in attendance for England's crunc...
4  9f6fbd3c497c4d28879bebebea220884f03eb41a  A Triple M Radio producer has been inundated w...  Nick Slater's colleagues uploaded a picture to...
2024-06-14 17:43:27,972 - INFO - Dataset preprocessed successfully
2024-06-14 17:43:27,972 - INFO -
Validation Dataset after cleaning and tokenization:
2024-06-14 17:43:27,972 - INFO -                                                 article                                      cleaned_text
             sentences
0  Sally Forrest, an actress-dancer who graced th...  sally forrest an actressdancer who graced the ...  [Sally Forrest, an actress-dancer who graced t...
1  A middle-school teacher in China has inked hun...  a middleschool teacher in china has inked hund...  [A middle-school teacher in China has inked hu...
2  A man convicted of killing the father and sist...  a man convicted of killing the father and sist...  [A man convicted of killing the father and sis...
3  Avid rugby fan Prince Harry could barely watch...  avid rugby fan prince harry could barely watch...  [Avid rugby fan Prince Harry could barely watc...
4  A Triple M Radio producer has been inundated w...  a triple m radio producer has been inundated w...  [A Triple M Radio producer has been inundated ...
2024-06-14 17:43:31,346 - INFO - Article length distribution plotted successfully
2024-06-14 17:43:34,521 - INFO - Sentence length distribution plotted successfully
2024-06-14 17:43:38,715 - INFO - Most common words plotted successfully
{'loss': 11.6499, 'grad_norm': 39.52219772338867, 'learning_rate': 1.0000000000000002e-06, 'epoch': 0.4}
{'loss': 10.737, 'grad_norm': 56.30098342895508, 'learning_rate': 2.0000000000000003e-06, 'epoch': 0.8}
{'train_runtime': 71.7995, 'train_samples_per_second': 1.393, 'train_steps_per_second': 0.348, 'train_loss': 11.271743774414062, 'epoch': 1.0}
100%|████████████████████████████████████████████████████████████████████████████| 25/25 [01:11<00:00,  2.87s/it]
```

- Train Runtime: 71.7995s
  Train Samples per second: 1.393
  Train Steps per second: 0.348
  Train Loss: 11.271
  epoch: 1.0

- ROGUE SCORES
  ROGUE-1: 0.3612
  ROGUE-2: 0.1448
  ROGUE-L: 0.2355