Infosys Internship 4.0 Project

Documentation

Title: TEXT SUMMARIZATION

By

NISHA

Mentor: Mr. Narendra Kumar

Contents

❖ Acknowledgment

- I would like to thank Infosys for the opportunity to intern as an AI/ML Intern .
- Special thanks to my mentor, Mr. Narendra Kumar, for his invaluable guidance and support.
- I am also grateful to my team and colleagues for their continuous support and collaboration throughout this project.
- Additionally, I am thankful to my family for their encouragement and assistance throughout this project.

❖ Problem Statement

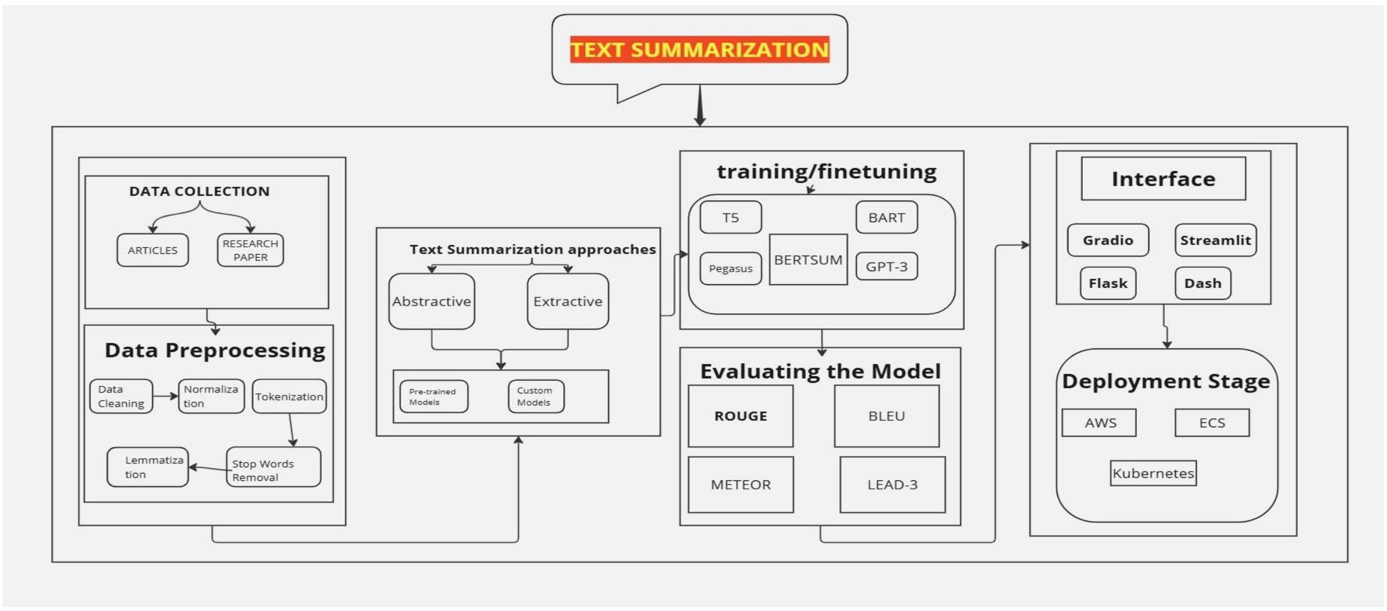
The project aims to develop an effective text summarization model that condenses extensive texts into concise summaries with high accuracy and relevance. By automatically generating informative summaries from long-form text, such as news articles, the model optimizes for accuracy, coherence, and relevance. Leveraging advanced natural language processing techniques, this initiative seeks to enhance information retrieval and readability across diverse domains.

❖ Introduction

In the digital age, the exponential growth of textual data has created a pressing need for efficient information processing toolsThis report presents the development of a text summarization system utilizing advanced natural language processing (NLP) techniques.The system employs two primary summarization methods: abstractive and extractive. Abstractive summarization generates new sentences that capture the essence of the original text, while extractive summarization selects key sentences directly from the source material. To implement these methods, the project leverages the T5-small model for abstractive summarization and the TextRank algorithm for extractive summarization.

This report outlines the problem statement, workflow, data collection, preprocessing steps, summarization methodologies, user interface design, and results. It also discusses observations, conclusions, and potential future enhancements, showcasing the system's ability to improve information retrieval and readability across various domains.

❖ workflow:



❖ Data collection

Data Source and Justification

- **Source**: I collected a dataset from the CNN/DailyMail website for this project. The CNN/DailyMail dataset is widely used in research due to its comprehensive news articles and corresponding summaries, making it particularly suited for text summarization tasks.
- **Relevance**: This dataset is ideal for training summarization models because of its rich and diverse content. It provides a robust basis for developing and testing text summarization algorithms.

Dataset Description

- **Initial Dataset**: The main dataset was initially saved as dataset.csv, containing 70,000 records.

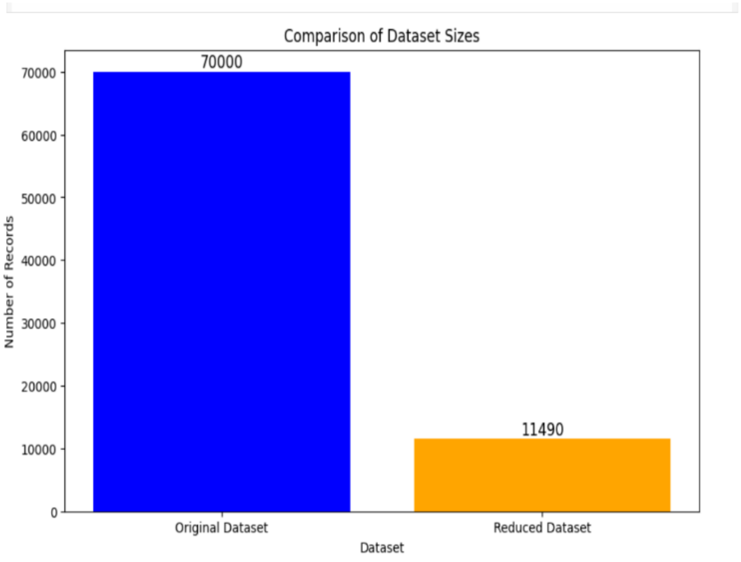
Structure:

- **id**: Unique identifier for each record.
- **article**: The body of the news article.
- **highlights**: The summary or highlights of the article.

Dataset Links

- **Master Dataset**: You can find the link to the master dataset here.

Data Reduction

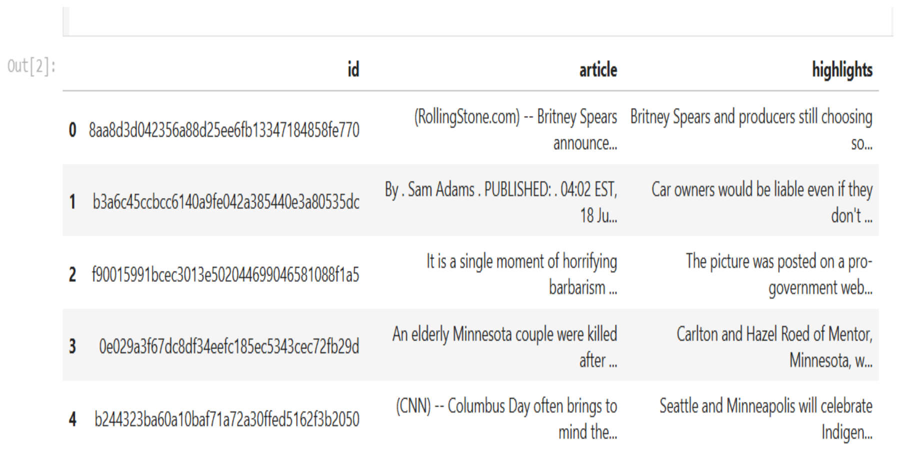


- **Reason for Reduction**: Due to constraints such as processing power, memory limitations, and the need for efficient handling, I reduced the dataset size to 11,490 records.
- **Sampling Method**: Random sampling was chosen to ensure that the reduced dataset remained representative of the original data distribution. This process helps in maintaining the diversity and richness of the dataset while making it manageable for processing and model training.
- Reduced Dataset: You can find the link to the reduced dataset here.

❖ Data Preprocessing

Objective:

- The primary goal of data preprocessing was to clean and prepare the dataset for model training by removing noise, tokenizing the text, and eliminating stopwords.
- The dataset before preprocessing is shown in the above picture.



Out[2]:

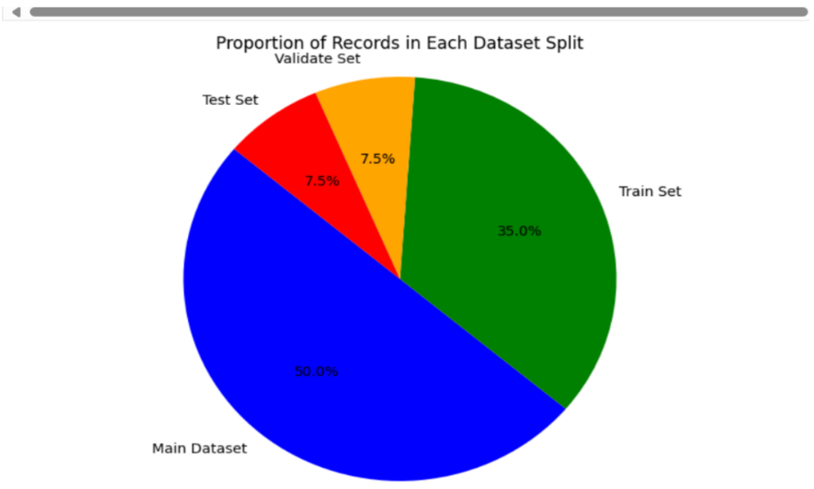
	id	article	highlights
0	8aa8d3d042356a88d25ee6fb13347184858fe770	(RollingStone.com) -- Britney Spears announce...	Britney Spears and producers still choosing so...
1	b3a6c45ccbcc6140a9fe042a385440e3a80535dc	By . Sam Adams . PUBLISHED: . 04:02 EST, 18 Ju...	Car owners would be liable even if they don't ...
2	f90015991bcec3013e502044699046581088f1a5	It is a single moment of horrifying barbarism ...	The picture was posted on a pro-government web...
3	0e029a3f67dc8df34eefc185ec5343cec72fb29d	An elderly Minnesota couple were killed after ...	Carlton and Hazel Roed of Mentor, Minnesota, w...
4	b244323ba60a10baf71a72a30ffed5162f3b2050	(CNN) -- Columbus Day often brings to mind the...	Seattle and Minneapolis will celebrate Indigen...

Steps Undertaken:

- **Cleaning Text Data:**
  - Removed special characters and unnecessary symbols using regular expressions to ensure the text contained only alphanumeric characters and spaces.
  - Converted all text to lowercase to maintain uniformity and consistency.
- **Tokenization:**
  - Split the cleaned text into individual words or tokens for more granular text analysis.
- **Stopword Removal:**
  - Identified and removed common stopwords (e.g., "the," "is," "in") using the NLTK library, which are not useful for understanding the content and context of the text.
- **Preprocessing Articles and Highlights:**

- Applied the above cleaning, tokenizing, and stopword removal steps to both the dataset's 'article' and 'highlights' columns.
- Ensured that both the news articles and their corresponding highlights were preprocessed uniformly.

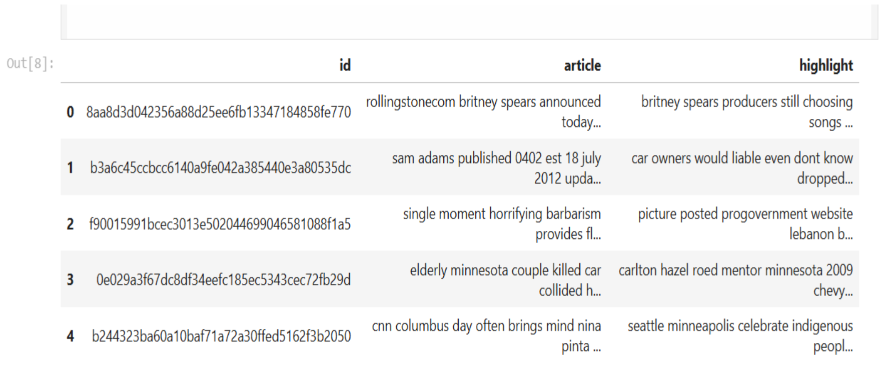
- **Dataset Storage:**
  - Stored the processed data in a new CSV file, demo.csv, maintaining the structure with 'id', 'article', and 'highlight' columns.
- **Dataset Splitting**:
  - Split the preprocessed dataset into three distinct sets: training (70%), validation (15%), and test (15%) sets to facilitate model training and evaluation.
  - A pie chart was generated to visually represent the proportion of records in each dataset split.



  - Ensured the splits were random but reproducible by setting a random state for consistency.
  - Saved each dataset split into separate CSV files (train_data1.csv, validate_data.csv, test_data.csv) for ease of access during the modeling phase.

Outcomes:

- Below picture shows cleaned dataset after preprocessing.
- The initial dataset of 11,490 records was successfully cleaned, tokenized, and filtered for stopwords.



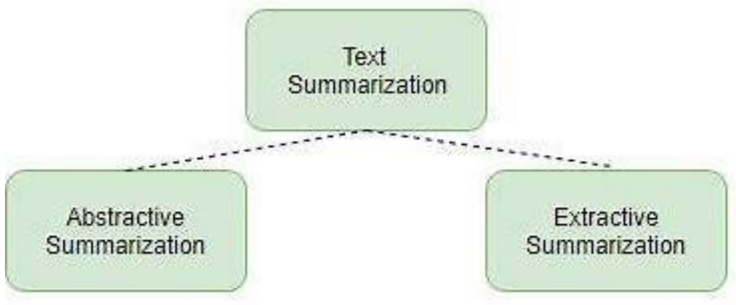
Out[8]:

	id	article	highlight
0	8aa8d3d042356a88d25ee6fb13347184858fe770	rollingstonecom britney spears announced today...	britney spears producers still choosing songs ...
1	b3a6c45ccbcc6140a9fe042a385440e3a80535dc	sam adams published 0402 est 18 july 2012 upda...	car owners would liable even dont know dropped...
2	f90015991bcec3013e5020446990465810888f1a5	single moment horrifying barbarism provides fl...	picture posted progovernment website lebanon b...
3	0e029a3f67dc8df34eefc185ec5343cec72fb29d	elderly minnesota couple killed car collided h...	carlton hazel roed mentor minnesota 2009 chevy...
4	b244323ba60a10baf71a72a30ffed5162f3b2050	cnn columbus day often brings mind nina pinta ...	seattle minneapolis celebrate indigenous peopl...

- The preprocessed dataset maintained the integrity of the original content while being optimized for text summarization tasks.
- The final dataset splits were appropriately proportioned, ensuring balanced representation across training, validation, and test sets.
- This setup is crucial for unbiased model evaluation and robust performance metrics.

➢ **Text summarization can generally be done in two main ways:**

- **1. Extractive Summarization:**
- **2. Abstractive Summarization:**



❖ Abstractive Summarization

- **Definition:** Abstractive summarization creates new sentences to capture the essence of the original text.

- **Method:** Uses advanced natural language processing techniques and models to understand context and generate concise summaries.
- **Examples:** T5, BART, and GPT.

Model training methodology

- **Training Approaches for Text Summarization Model**

1. **Training from Scratch:** Building a model from the ground up using a large dataset.
2. **Pre-trained Models:** Utilizing pre-trained models and fine-tuning them on specific datasets.

- **Reason for Selecting Pre-trained Models**
    - **Efficiency:** Pre-trained models require less data and time for training
    - **Performance:** Achieve higher accuracy and better performance metrics.
- **Selected model:** T5-small Transformer model from Hugging Face
- **Reason for Selecting T5-small:**
    - Strong balance between computational efficiency and performance, making it an ideal choice for text summarization projects.
- **Training process:**
    - **Data Preparation:**
        1. Tokenization: T5Tokenizer for input text and target summaries
        2. Custom dataset class (MyDataset):
        3. PyTorch DataLoader:
    - **Hyperparameters:**

1. Batch size: Training and validation datasets were processed.
2. AdamW optimizer: used for model optimization.
3. Maximum Lengths: Maximum input length was set to 512 tokens, and output length to 150 tokens.

- **Training Loop:**
  1. Epochs: Multiple with forward and backward passes per epoch.
  2. Loss Computation:
  3. Average Loss Calculation:
- **Evaluation and Early Stopping:**

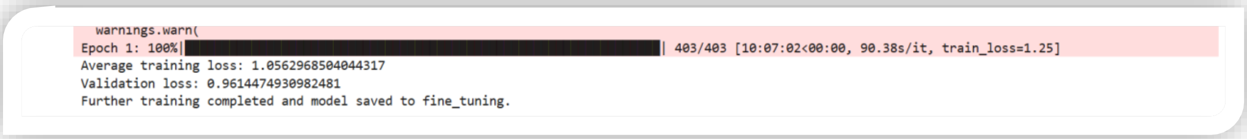
1. Validation Loss: Computed after each epoch.
2. Early Stopping: Patience of 3 epochs to prevent overfitting.

- **Model Saving:**

1. Best Model: Saved based on validation loss improvement.

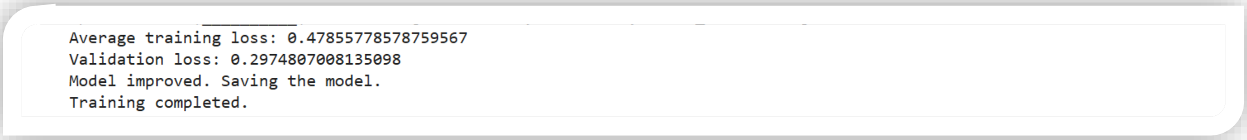
- **Hyperparameter Tuning:**

1. Adjustments: Fine-tuned hyperparameters such as learning rate and number of epochs.



```
  warnings.warn(
Epoch 1: 100%|██████████████████████████████████████████████████| 403/403 [10:07:02<00:00, 90.38s/it, train_loss=1.25]
Average training loss: 1.0562968504044317
Validation loss: 0.9614474930982481
Further training completed and model saved to fine_tuning.
```

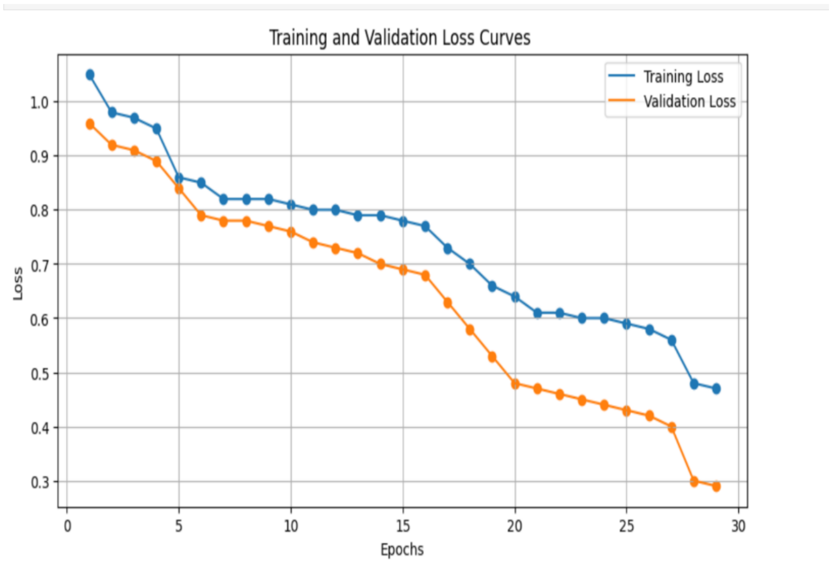
2. Initial Losses: Training loss started at 1.05 and validation loss at 0.96.



```
Average training loss: 0.47855778578759567
Validation loss: 0.2974807008135098
Model improved. Saving the model.
Training completed.
```

3. Final Losses: Reduced to 0.47 for training and 0.29 for validation.

- **Observations:**
  - **Loss Reduction Post Hyperparameter Tuning:**



1. The validation loss saw a significant decrease from an initial 0.96 to a final 0.29 after hyperparameter tuning.
2. Simultaneously, the training loss also reduced notably from 1.05 to 0.40, indicating improved model convergence and effectiveness in generating accurate summaries

❖ Performance Matrix

- **Objective:**
  - **Evaluate Model Performance:** Quantitatively assess how well the generated summaries match the reference summaries.

- **Available Evaluation Metrics:**
  - BLEU (Bilingual Evaluation Understudy):
  - METEOR (Metric for Evaluation of Translation with Explicit ORdering):

- ROUGE (Recall-Oriented Understudy for Gisting Evaluation):

- **Used Performance Matrix (ROUGE):**

  - **ROUGE Score (**Recall-Oriented Understudy for Gisting Evaluation).
  - **Reason:** comprehensive evaluation of summary quality across different n-gram overlaps, providing a robust benchmark for model assessment.

- **ROUGE Score Analysis:**

1. **ROUGE-1:**
   - **Unigram Overlap:** Indicates presence of relevant words.
2. **ROUGE-2:**
   - **Bigram Overlap:** Reflects fluency and context.
3. **ROUGE-L:**
   - **Longest Common Subsequence:** Assesses structure and sequence.

❖ Implementation:

○ Importing Libraries:

1. **pandas:** For data manipulation.
2. **torch:** For model handling.
3. **transformers:** For tokenization and model loading.
4. **tqdm:** For progress tracking.
5. **rouge_score:** For calculating ROUGE scores.

○ Dataset Class Definition:

- **MyDataset Class:**

1. **Initialization:** Takes data, tokenizer, and specified max lengths.
2. **Length Method:** Returns the dataset size.
3. **Get Item Method:** Preprocesses individual data points.

- **Calculating ROUGE Scores:**

  - **Function:** calculate_rouge_scores(hypotheses, references)

1. **Initialization:** Sets up a ROUGE scorer.
2. **Iteration:** Computes individual ROUGE scores.
3. **Averaging:** Calculates average ROUGE-1, ROUGE-2, and ROUGE-L scores.

- **Loading and Preprocessing Validation Data:**

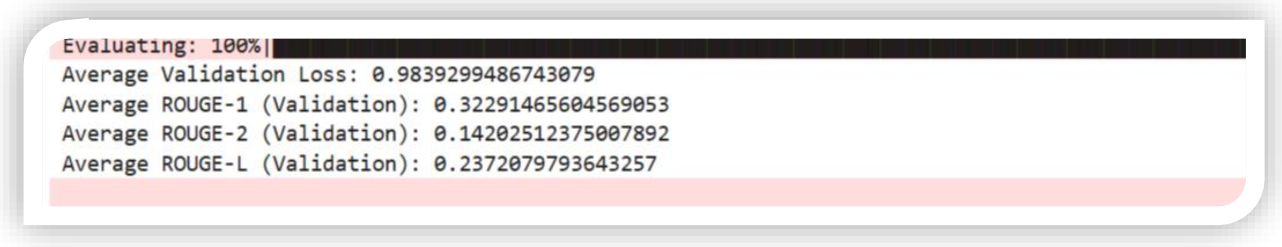
1. **Load Data:** Reads validation data from CSV.
2. **Initialize Tokenizer and Model:** Uses pre trained T5 model.
3. **Create DataLoader:** For validation dataset.

- **Model Evaluation:**

1. **Set Device:** Uses GPU if available.
2. **Evaluation Mode:** Sets model to eval mode.
3. **Lists for Summaries:** Stores generated and target summaries.
4. **Inference:** Generates summaries and decodes them.

- **Calculating and Printing ROUGE Scores:**

1. **Call Function:** calculate_rouge_scores.

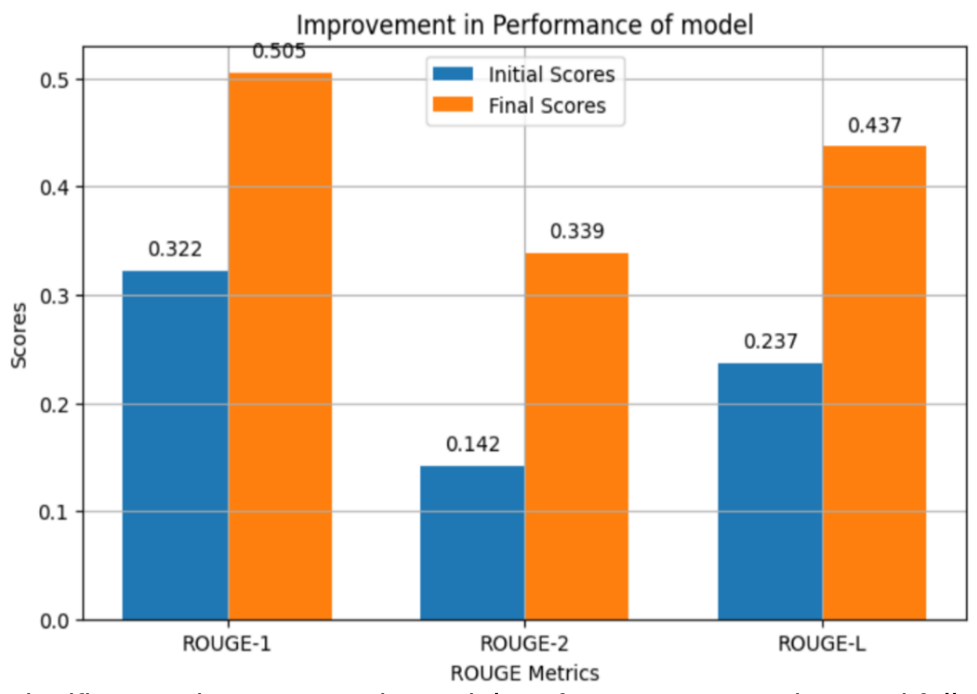


```
Evaluating: 100%|
Average Validation Loss: 0.9839299486743079
Average ROUGE-1 (Validation): 0.32291465604569053
Average ROUGE-2 (Validation): 0.14202512375007892
Average ROUGE-L (Validation): 0.2372079793643257
```

2. **Print Scores:** Displays ROUGE-1, ROUGE-2, and ROUGE-L.

- **Observations:**

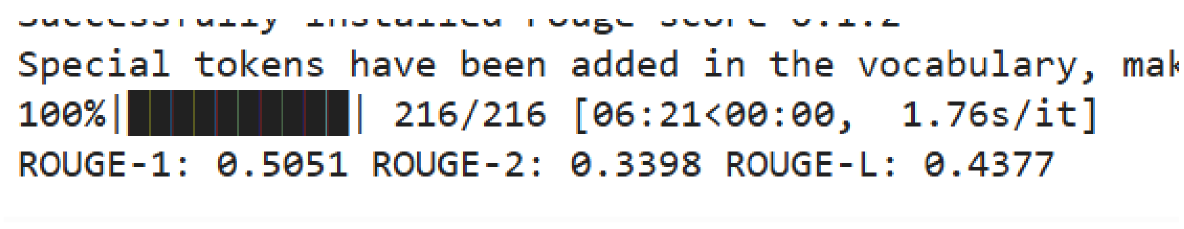
1. **Performance Improvement:**



- Significant enhancements in model performance were observed following rigorous hyperparameter tuning.
- Initial ROUGE-1, ROUGE-2, and ROUGE-L scores stood at 0.322, 0.142, and 0.237 respectively.
- Post tuning, these metrics improved to 0.505, 0.339, and 0.437, demonstrating substantial progress in summarization accuracy.

2. **Impact of Hyperparameter Tuning:**

- The optimized configuration of hyperparameters played a pivotal role in refining the model's architecture and training process.
- This optimization led to better alignment between generated summaries and reference texts, as evidenced by the enhanced ROUGE scores.



```
Successfully installed rouge-score 0.1.2
Special tokens have been added in the vocabulary, mak
100%|██████████| 216/216 [06:21<00:00,  1.76s/it]
ROUGE-1: 0.5051 ROUGE-2: 0.3398 ROUGE-L: 0.4377
```

❖ Extractive text summarization

- **Objective:** The objective is to develop a system that generates concise summaries by selecting key sentences from the source text, and preserving original wording and context.
- **Method:** Ranks and extracts the most important sentences from the original text.
- **Examples:** Techniques like TextRank, TF-IDF, and pre-trained models such as BERT and GPT.

Model training methodology

- **Used textrank algorithm**
  - Instead of choosing computationally intensive deep learning models, I used the TextRank algorithm, which provides an optimal solution through a graph-based approach.
- **Reason:**

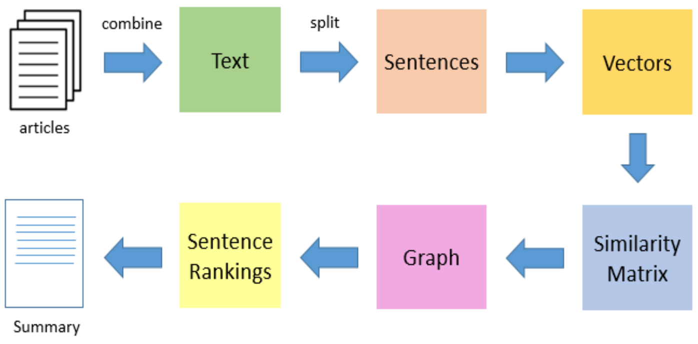
- TextRank was selected for extractive summarization due to its simplicity, interpretability, resource efficiency, and ability to maintain high fidelity to the source text.
- It is based on the concept that words which occur more frequently are significant.Based on this, the algorithm assigns scores to each sentence in the text. The top-ranked sentences make it to the summary.

- **Methodology**

1. Dataset Preparation and Preprocessing

- Dataset Acquisition: Import the dataset into a Pandas DataFrame for efficient handling.

- Text Preprocessing:
  - Tokenization: Use NLTK's tokenizer to segment text into sentences and words.



  - Stopword Removal: Remove common and non-informative words using NLTK's stopwords.

2. **Building the Similarity Matrix**

- Sentence Similarity Calculation:
  - Cosine Similarity: Calculate the similarity between sentences based on the cosine similarity of word frequency vectors.
- Constructing the Matrix:

        - Matrix Representation: Create a square matrix where each cell represents the similarity score between two sentences.

3. **TextRank Algorithm Implementation**
    - Graph Representation:
        - Graph Construction: Transform the similarity matrix into a graph using NetworkX, where sentences are nodes and similarity scores are edges.
    - PageRank Computation:
        - Ranking Sentences: Apply the PageRank algorithm to assign importance scores to sentences based on their connectivity and similarity within the graph.
4. **Summary Generation:**

- Top N Sentences: Select the top N sentences with the highest PageRank scores to form the summary.

- **Performance Metrics and Scores**

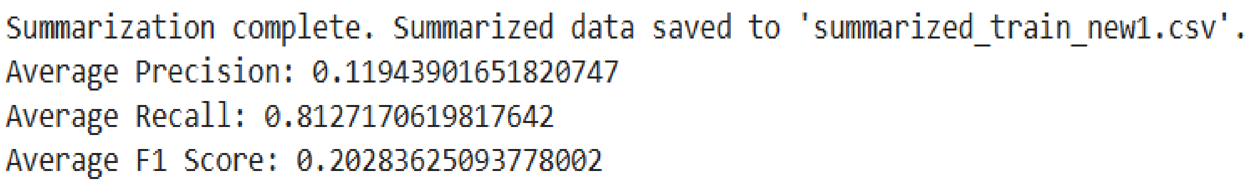
- **Used Performance Matrix (ROUGE):**
    - ROUGE Score **(**Recall-Oriented Understudy for Gisting Evaluation).

- **Methodology**
    - Evaluation Metrics:

- Precision: Ratio of correctly extracted sentences to the total sentences in the generated summary.
- Recall: Completeness of the generated summary compared to the reference summary.
- F1 Score: Harmonic mean of precision and recall, providing a balanced measure of summarization quality.

5. **Implementation Details:**
    - Iterative Processing: Each article undergoes summarization using the TextRank algorithm.
    - Comparison with Reference Summaries: Generated summaries are compared against reference summaries (stored in the 'highlight' column) to compute precision, recall, and F1 scores.

- **RESULT**:
    - Precision: The average precision score was approximately 0.119.
    - Recall: The average recall score was approximately 0.813.
    - F1 Score: The average F1 score was approximately 0.203.



```
Summarization complete. Summarized data saved to 'summarized_train_new1.csv'.
Average Precision: 0.11943901651820747
Average Recall: 0.8127170619817642
Average F1 Score: 0.20283625093778002
```

- **Observations**

    - **High Recall:** High recall indicates that the model captures most of the relevant information from the original text.

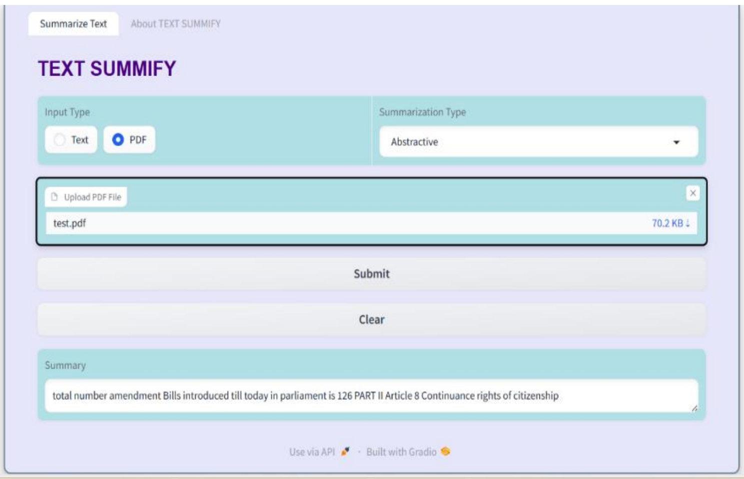
    - **Moderate Precision and F1 Scores:** The model shows moderate precision (0.119) and a balanced F1 score (0.203), indicating the potential for improving sentence relevance selection in summaries.
    - **Resource Efficiency:** The model is resource-efficient and suitable for environments with limited computational power.

❖ User Interface

- **Overview**
    - The interface is developed using the Gradio library, providing an interactive platform for text summarization.
    - Users can input text or upload PDF files and choose between abstractive and extractive summarization methods.
- **Reason**: Gradio is used to create the interface because it provides a simple and intuitive platform for building interactive web applications with minimal code.
- **Interface Components**
    - Input Type Selection**:** Radio button to choose text input or PDF upload.
    - Summarization Type Selection: Dropdown menu for abstractive or extractive summarization.
    - Text Input Area: Textbox for entering text.
    - PDF File Upload: Button to upload a PDF document.
    - Submit Button**:** Initiates the summarization process.
    - Clear Button**:** Clears input fields and output area.
    - Output Area**:** Displays the summarized text.

- Description Page: Markdown page with application information and usage instructions.

- **Implementation Overview**
  - Abstractive Summarization Function: Uses the T5 model to generate summaries by encoding the input text and decoding the generated summary.
  - Extractive Summarization Function: Implements the TextRank algorithm using CountVectorizer, cosine similarity, and PageRank to rank and select sentences for the summary.
  - PDF Text Extraction: Uses PyMuPDF (fitz) to extract text from PDF files.
  - Summarization Logic: Determines input type (text or PDF) and summarization type (abstractive or extractive) to process the input and generate the summary.



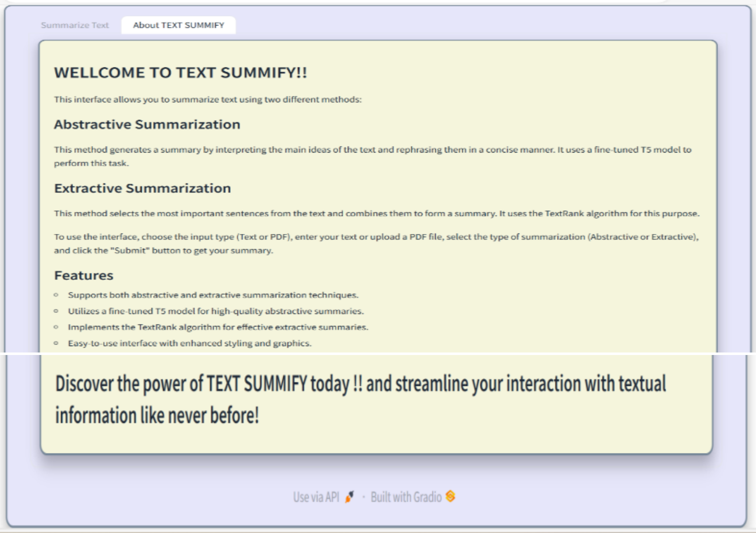
Input Type: Text [✓ PDF]

Summarization Type: Abstractive

Upload PDF File: test.pdf 70.2 KB

Summary: total number amendment Bills introduced till today in parliament is 126 PART II Article 8 Continuance rights of citizenship

  - Gradio Interface: Built with radio buttons, dropdowns, textboxes, file upload, and buttons for submitting and clearing inputs. A description page provides user guidance.
  - Styling: Custom CSS for enhancing the visual appearance of the interface.



WELLCOME TO TEXT SUMMIFY!!

This interface allows you to summarize text using two different methods:

Abstractive Summarization

This method generates a summary by interpreting the main ideas of the text and rephrasing them in a concise manner. It uses a fine-tuned T5 model to perform this task.

Extractive Summarization

This method selects the most important sentences from the text and combines them to form a summary. It uses the TextRank algorithm for this purpose.

To use the interface, choose the input type (Text or PDF), enter your text or upload a PDF file, select the type of summarization (Abstractive or Extractive), and click the "Submit" button to get your summary.

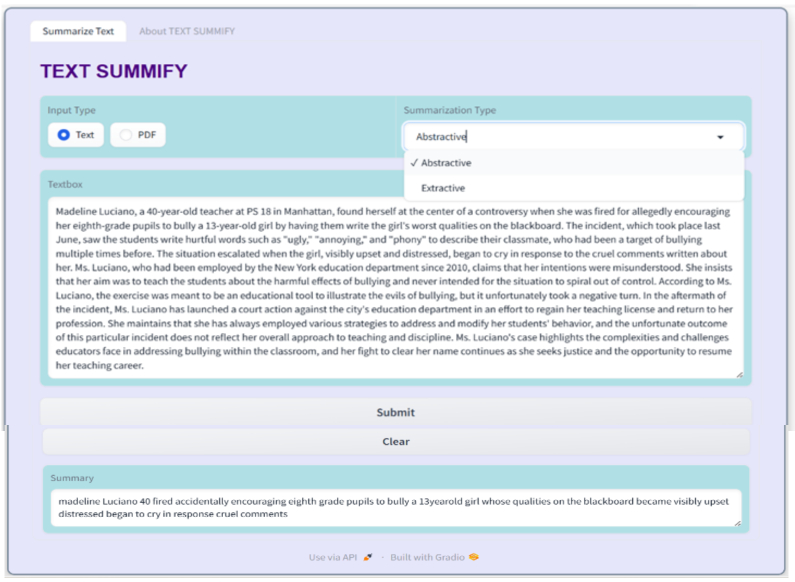
Features

- Supports both abstractive and extractive summarization techniques.
- Utilizes a fine-tuned T5 model for high-quality abstractive summaries.
- Implements the TextRank algorithm for effective extractive summaries.
- Easy-to-use interface with enhanced styling and graphics.

Discover the power of TEXT SUMMIFY today !! and streamline your interaction with textual information like never before!

Fig: description page

❖ results


Summarize Text | About TEXT SUMMIFY

TEXT SUMMIFY

Input Type: [✓ Text] PDF

Summarization Type: Abstractive

✓ Abstractive

Extractive

Textbox

Madeline Luciano, a 40-year-old teacher at PS 18 in Manhattan, found herself at the center of a controversy when she was fired for allegedly encouraging her eighth-grade pupils to bully a 13-year-old girl by having them write the girl's worst qualities on the blackboard. The incident, which took place last June, saw the students write hurtful words such as "ugly," "annoying," and "phony" to describe their classmate, who had been a target of bullying multiple times before. The situation escalated when the girl, visibly upset and distressed, began to cry in response to the cruel comments written about her. Ms. Luciano, who had been employed by the New York education department since 2010, claims that her intentions were misunderstood. She insists that her aim was to teach the students about the harmful effects of bullying and never intended for the situation to spiral out of control. According to Ms. Luciano, the exercise was meant to be an educational tool to illustrate the evils of bullying, but it unfortunately took a negative turn. In the aftermath of the incident, Ms. Luciano has launched a court action against the city's education department in an effort to regain her teaching license and return to her profession. She maintains that she has always employed various strategies to address and modify her students' behavior, and the unfortunate outcome of this particular incident does not reflect her overall approach to teaching and discipline. Ms. Luciano's case highlights the complexities and challenges educators face in addressing bullying within the classroom, and her fight to clear her name continues as she seeks justice and the opportunity to resume her teaching career.

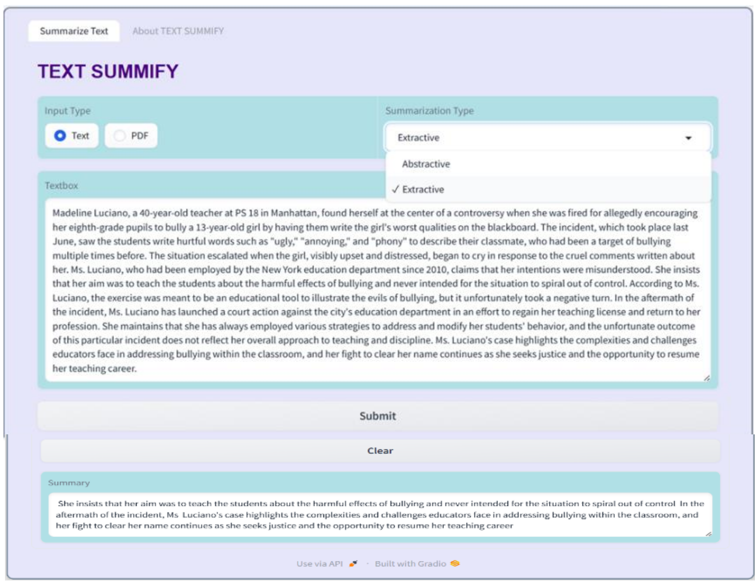
Submit

Clear

Summary

madeline Luciano 40 fired accidentally encouraging eighth grade pupils to bully a 13yearold girl whose qualities on the blackboard became visibly upset distressed began to cry in response cruel comments

Use via API · Built with Gradio


Summarize Text | About TEXT SUMMIFY

TEXT SUMMIFY

Input Type: [✓ Text] PDF

Summarization Type: Extractive

Abstractive

✓ Extractive

Textbox

Madeline Luciano, a 40-year-old teacher at PS 18 in Manhattan, found herself at the center of a controversy when she was fired for allegedly encouraging her eighth-grade pupils to bully a 13-year-old girl by having them write the girl's worst qualities on the blackboard. The incident, which took place last June, saw the students write hurtful words such as "ugly," "annoying," and "phony" to describe their classmate, who had been a target of bullying multiple times before. The situation escalated when the girl, visibly upset and distressed, began to cry in response to the cruel comments written about her. Ms. Luciano, who had been employed by the New York education department since 2010, claims that her intentions were misunderstood. She insists that her aim was to teach the students about the harmful effects of bullying and never intended for the situation to spiral out of control. According to Ms. Luciano, the exercise was meant to be an educational tool to illustrate the evils of bullying, but it unfortunately took a negative turn. In the aftermath of the incident, Ms. Luciano has launched a court action against the city's education department in an effort to regain her teaching license and return to her profession. She maintains that she has always employed various strategies to address and modify her students' behavior, and the unfortunate outcome of this particular incident does not reflect her overall approach to teaching and discipline. Ms. Luciano's case highlights the complexities and challenges educators face in addressing bullying within the classroom, and her fight to clear her name continues as she seeks justice and the opportunity to resume her teaching career.

Submit

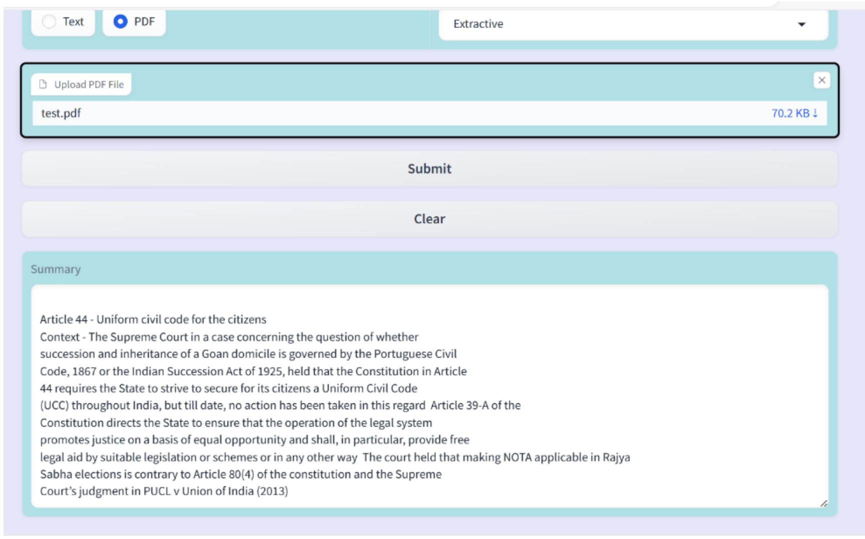
Clear

Summary

She insists that her aim was to teach the students about the harmful effects of bullying and never intended for the situation to spiral out of control In the aftermath of the incident, Ms Luciano's case highlights the complexities and challenges educators face in addressing bullying within the classroom, and her fight to clear her name continues as she seeks justice and the opportunity to resume her teaching career

Use via API · Built with Gradio

- The above figure shows the interface that summarizes long text into a summary using abstractive summarization with the T5-small model and extractive summarization using text rank.
- the Interface also summarizes PDF files by extracting text using PyMuPDF (fitz).


Text [✓ PDF]

Extractive

Upload PDF File

test.pdf 70.2 KB ↓

Submit

Clear

Summary

Article 44 - Uniform civil code for the citizens
Context - The Supreme Court in a case concerning the question of whether
succession and inheritance of a Goan domicile is governed by the Portuguese Civil
Code, 1867 or the Indian Succession Act of 1925, held that the Constitution in Article
44 requires the State to strive to secure for its citizens a Uniform Civil Code
(UCC) throughout India, but till date, no action has been taken in this regard Article 39-A of the
Constitution directs the State to ensure that the operation of the legal system
promotes justice on a basis of equal opportunity and shall, in particular, provide free
legal aid by suitable legislation or schemes or in any other way The court held that making NOTA applicable in Rajya
Sabha elections is contrary to Article 80(4) of the constitution and the Supreme
Court's judgment in PUCL v Union of India (2013)

- **Observations**
  - Efficient performance in summarizing both text and PDF inputs.
  - It produced accurate summaries using the T5 model for abstractive summarization and TextRank for extractive summarization.
  - Summarized outputs were concise and preserved essential information from the original texts or documents.

❖ Challenges Faced and Solutions

Challenge: Initially faced difficulties in improving model performance due to resource limitations and increased training times on local systems.

Solution: Overcame these limitations by migrating computations to Google Colab, utilizing powerful GPUs to expedite training and inference processes. Additionally, downsized the dataset through strategic sampling and preprocessing techniques to optimize efficiency without compromising quality.

❖ Conclusion

- The development of an automated text summarization system using advanced NLP techniques has proven effective and efficient.
- The Gradio interface provided a user-friendly platform for summarizing long texts and PDF documents.
- Using the T5-small model for abstractive summarization and the TextRank algorithm for extractive summarization, the system produced concise, relevant summaries that retained essential information.
- This project highlighted the potential of NLP technologies to enhance information retrieval and readability, benefiting various business applications and beyond.

❖ Future scope

- Model Improvement: Implementing more advanced pre-trained models such as T5-base or T5-large could further improve the accuracy and coherence of the summaries.

- Multi-language Support: Expanding the system to support multiple languages to cater to a more diverse user base.
- Additional Summarization Techniques: Exploring and incorporating other summarization methods, such as neural network-based extractive summarization and hybrid models, to offer users more choices.
- Integration with Other Platforms: Integrating the summarization tool with popular content management systems and collaboration platforms to broaden its applicability.