A NLP report on

TEXT SUMMARIZATION

Submitted by:

A.Niharika Varma-100521729005
Ch.ManasaGangotri-100521729014
L.V.S.Lalitha-100521729035
Sai Chandana-100521729050

Under the guidance of

Dr. S PADMAJA

Associate Professor, Department of CSE

KMIT, Narayanguda - 5000029

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING (AIML)

UNIVERSITY COLLEGE OF ENGINEERING

Osmania University, Hyderabad 3-5-1206, Narayanguda, Hyderabad – 500029

2023-2024

Index

List of Images

Introduction

1.1 Purpose of the Project

The primary purpose of this project is to develop an efficient and effective text summarization system utilizing both abstractive and extractive methodologies. With the exponential growth of digital information, the ability to quickly extract meaningful summaries from large volumes of text is crucial. This project aims to address the challenge by leveraging natural language processing (NLP) models to generate concise, coherent, and contextually accurate summaries. By employing both abstractive and extractive techniques, we seek to explore the strengths and limitations of each approach and provide a comprehensive solution that enhances information accessibility and usability.

1.2 Problem with the Existing System

1. Quality and Coherence: Many current summarization systems produce outputs that lack coherence and context, making them difficult to understand. This is particularly true for extractive methods, which may concatenate disjointed sentences without regard to overall readability.
2. Contextual Understanding: Traditional extractive methods rely heavily on selecting sentences directly from the source text, which can lead to summaries that miss the nuanced meaning or context intended by the original author.
3. Scalability and Efficiency: As the volume of available information grows, the scalability and efficiency of summarization systems become increasingly important. Existing models may not be capable of handling large datasets or generating summaries in a timely manner.
4. Diversity of Applications: Current systems may not be versatile enough to handle different types of texts, such as news articles, scientific papers, or social media posts, each of which may require different summarization strategies.

5. Evaluation Metrics: Assessing the quality of summaries remains a challenge, with many existing systems relying on metrics that do not fully capture human judgment or the multifaceted nature of a good summary.

1.3 Proposed System

To address the shortcomings of existing systems, we propose a hybrid text summarization approach that combines both abstractive and extractive methods. Our system utilizes several cutting-edge models:

- T5 Model: This model (Text-to-Text Transfer Transformer) converts text inputs to text outputs, allowing for highly flexible and accurate summarizations. It excels in generating human-like text that maintains the original context.
- Pegasus Model: Specifically designed for summarization, Pegasus uses a pre-training objective that masks and generates important sentences, resulting in high-quality summaries that capture the essence of the source text.
- BERT-based Model: Utilizing transformers for text generation, this model provides a robust approach to understanding and summarizing complex texts.
- LexRank Model: Based on graph-based algorithms, LexRank identifies the most salient sentences in a text, ensuring that the most important information is included in the summary.
- Latent Semantic Analysis (LSA): This method uses singular value decomposition to identify key concepts and relationships within the text, selecting sentences that best represent these concepts.
- TextRank Model: An unsupervised algorithm inspired by PageRank, TextRank constructs a graph of sentences and uses ranking algorithms to determine the significance of each sentence, extracting the most relevant ones for the summary.
- TF-IDF: Term Frequency-Inverse Document Frequency (TF-IDF) scores words based on their importance in a document relative to a corpus. Sentences with the highest TF-IDF scores are selected to ensure that the summary covers the most critical topics.

Our proposed system aims to leverage the strengths of both approaches, offering a comprehensive solution that produces high-quality summaries across various types of texts. Additionally, the system incorporates advanced evaluation metrics to ensure that the summaries are both accurate and coherent.

1.4 Scope of the Project

The scope of this project encompasses several key objectives:

1. Model Development and Training: Develop and fine-tune the T5, Pegasus, and BERT-based models for abstractive summarization, and implement the LexRank and LSA models for extractive summarization.
2. Dataset Collection and Preprocessing: Gather a diverse set of texts from various domains (e.g., news articles, academic papers, social media posts) and preprocess them to ensure compatibility with the models.
3. Performance Evaluation: Implement robust evaluation metrics, including ROUGE scores, human evaluation, and readability assessments, to measure the quality of the generated summaries.
4. User Interface Design: Develop a user-friendly interface that allows users to input text and receive summaries in real-time, with options to customize the summarization settings.
5. Scalability and Efficiency: Ensure that the system can handle large volumes of text and generate summaries in a timely manner, suitable for both individual and enterprise use.

1.5 Architecture Diagram

The architecture of our text summarization system is designed to evaluate the accuracy and effectiveness of various summarization models, both abstractive and extractive methods. The following diagram outlines the key components and workflow of the system:

Architecture Diagram Components

1. **Input Layer**: Users input the text to be summarized through a web interface or API.
2. **Preprocessing Module**: The input text undergoes preprocessing steps such as tokenization, normalization, and removal of irrelevant content to prepare it for summarization.
3. **Summarization Module**:
   - **Abstractive Models**: T5, Pegasus, and BERT-based models generate summaries by understanding and rephrasing the content.
   - **Extractive Models**: LexRank, LSA, TextRank, and TF-IDF models extract key sentences from the text based on their significance and relevance.
4. **Evaluation Module**: This module evaluates the summaries generated by different models using metrics such as ROUGE scores, coherence, and human judgment.
5. **Output Layer**: The final summaries, along with their evaluation metrics, are presented to the user through the interface.

2. Software Requirements Specifications

2.1 Requirements Specification Document

The Requirements Specification Document outlines the technical and functional specifications needed to develop, implement, and maintain the text summarization system. It serves as a comprehensive guide for the development team, ensuring all aspects of the project are thoroughly planned and documented.

2.2 Functional Requirements

Functional requirements describe the specific behaviors and functions the system must exhibit.

1. **User Input Handling**:
   - The system must accept text input from various sources.
   - It should support different text formats (e.g., plain text, PDF, DOCX).
2. **Preprocessing**:

   - The system must preprocess input text by tokenizing, normalizing, and removing irrelevant content.

3. **Summarization**:
   - The system must implement both abstractive and extractive summarization techniques.
   - It should integrate T5, Pegasus, BERT-based models for abstractive summarization.
   - It should include LexRank, LSA, TextRank, and TF-IDF for extractive summarization.
4. **Output Generation**:
   - The system must generate summaries that are coherent, concise, and contextually accurate.
5. **User Interface**:
   - The system must provide a user-friendly interface for text input and summary output.

2.3 Non-Functional Requirements

Non-functional requirements specify the quality attributes of the system. These include:

1. **Performance**:
   - The system must generate summaries within an acceptable timeframe (e.g., less than 5 seconds for typical text inputs).
   - It should efficiently handle large volumes of text.
2. **Scalability**:
   - The system must be scalable to accommodate increasing numbers of users and larger datasets.
3. **Reliability**:
   - The system must be reliable, providing consistent and accurate summaries without failures.
4. **Usability**:

  - The user interface must be intuitive and easy to navigate, requiring minimal user training.
5. **Maintainability**:
   - The system must be easy to maintain and update, with clear documentation and modular components.

2.4 Software Requirements

The software requirements detail the necessary software components and dependencies needed to build and run the system:

1. **Operating System**:
   - Linux, Windows, or macOS.
2. **Programming Languages**:
   - Python for the main development of NLP models and backend processes.
3. **Frameworks and Libraries**:
   - TensorFlow and PyTorch for deep learning models.
   - Transformers library from Hugging Face for implementing T5, Pegasus, and BERT-based models.
   - NLTK, SpaCy, and Gensim for natural language processing tasks.
   - NetworkX for graph-based algorithms like LexRank and TextRank.
   - Scikit-learn for implementing TF-IDF.
4. **Web Framework**:
   - Flask or Django for developing the web interface.
5. **APIs**:
   - RESTful APIs for integrating with other applications and services.

2.5 Hardware Requirements

Development and Testing:

- Multi-core processor (Intel i5/i7 or AMD Ryzen 5/7)
- 16 GB RAM

- Dedicated GPU (NVIDIA GTX 1080 or higher)
- 512 GB SSD storage

Production Deployment:

- Multi-core server CPUs (Intel Xeon or AMD EPYC)
- 32-64 GB RAM
- Multiple GPUs (NVIDIA Tesla V100 or A100)

Backup and Redundancy:

- Load balancing and redundancy servers
- Regular backup systems for data recovery

3. Literature Survey

A literature survey provides a comprehensive overview of existing research and methodologies in text summarization, helping to contextualize our project within the current landscape.

3.1 Understanding the State of the Art

- **Abstractive Summarization**: Examines transformer-based models like T5, Pegasus, and BERT. These models generate human-like summaries by rephrasing content and capturing contextual meaning.
- **Extractive Summarization**: Discusses algorithms such as LexRank, LSA, TextRank, and TF-IDF. These methods identify and extract the most significant sentences from a text.

3.2 Identifying Gaps and Challenges

- **Coherence and Context**: Many existing systems struggle with producing coherent and contextually accurate summaries.
- **Scalability**: Handling large datasets efficiently remains a significant challenge.

- **Evaluation Metrics**: There is a need for more comprehensive metrics that capture human judgment effectively.

3.3 Establishing the Theoretical Framework

- **Abstractive Models**: Reviews the principles behind transformer architectures and their application in summarization.
- **Extractive Models**: Explores graph-based and statistical methods for sentence extraction.

3.4 Informing Methodology

- **Model Selection**: Insights from the survey guide the choice of models for our project, such as T5 and Pegasus for abstractive summarization, and LexRank and TF-IDF for extractive summarization.
- **Techniques**: The survey informs preprocessing and evaluation techniques, ensuring robust and effective summarization.

3.5 Benchmarking and Evaluation

- **Evaluation Metrics**: Discusses metrics like ROUGE, essential for assessing summary quality.
- **Benchmark Studies**: Provides performance benchmarks from previous studies to compare our results against established standards.

4. Model Building

In this section, we provide a comprehensive overview of the processes involved in building and implementing both abstractive and extractive summarization models. Each stage of model building is crucial for ensuring high-quality, accurate summaries.

4.1 Data Preprocessing

Data preprocessing prepares the input text for summarization by transforming it into a format suitable for the models.

1. **Tokenization**:

  - The process of breaking down the text into individual words or subwords. This step is essential for both abstractive and extractive models.
  - Tools used: NLTK, SpaCy, and Hugging Face Tokenizers.

2. **Normalization**:
   - Converting text to a consistent format, such as lowercasing, removing punctuation, and handling special characters.
   - This step ensures uniformity and reduces variability in the text data.
3. **Stopword Removal**:
   - Eliminating common words that do not contribute significant meaning, such as "the," "is," and "and."
   - This step helps in focusing on the meaningful content of the text.
4. **Sentence Splitting**:
   - Dividing the text into individual sentences, which is particularly important for extractive summarization models.
   - Tools used: NLTK's sentence tokenizer, SpaCy.
5. **Lemmatization and Stemming**:
   - Converting words to their base or root form to reduce inflectional forms and derivationally related forms.
   - Tools used: NLTK, SpaCy.

1. **T5 Model (Text-to-Text Transfer Transformer)**:
   - Converts all NLP tasks into a text-to-text format, which provides a unified approach to handling various tasks, including summarization.
   - **Training**: Pre-trained on a large corpus and fine-tuned on a summarization-specific dataset.
   - **Advantages**: Versatility and strong performance across different NLP tasks.
   - **Implementation**: Using Hugging Face's Transformers library for model loading and fine-tuning.
2. **Pegasus Model**:

- Specifically designed for summarization, using a pre-training objective that masks and generates important sentences.
- **Training**: Pre-trained on a large summarization corpus and fine-tuned on domain-specific data.
- **Advantages**: Superior performance in capturing key points and generating coherent summaries.
- **Implementation**: Leveraging TensorFlow or PyTorch with the Transformers library.

3. **BERT-based Model**:
    - Utilizes the BERT architecture to deeply understand context and generate human-like summaries.
    - **Training**: Fine-tuned for the summarization task, using a dataset specifically designed for this purpose.
    - **Advantages**: Strong contextual understanding and flexibility.
    - **Implementation**: Using the Hugging Face library for model fine-tuning and inference.

1. **Latent Semantic Analysis (LSA)**:
    - Uses singular value decomposition (SVD) to capture underlying concepts and relationships within the text.
    - **Algorithm**: Decomposes the term-document matrix to identify patterns and select representative sentences.
    - **Advantages**: Captures semantic structure well.
    - **Implementation**: Using Scikit-learn’s TruncatedSVD and NLTK.
2. **TextRank Model**:
    - Inspired by Google's PageRank, constructs a graph of sentences and ranks them based on their relevance.
    - **Algorithm**: Uses sentence co-occurrence and importance to rank sentences.
    - **Advantages**: Unsupervised and effective for various types of text.
    - **Implementation**: Using Gensim and NetworkX libraries.
3. **TF-IDF**:

  - Scores sentences based on term frequency-inverse document frequency, highlighting important words in the context of the document.
  - **Algorithm**: Calculates TF-IDF scores for words and selects sentences with the highest cumulative scores.
  - **Advantages**: Simple and interpretable.
  - **Implementation**: Using Scikit-learn's TfidfVectorizer.

4.4 Model Training

Training the models involves multiple steps to ensure optimal performance and accuracy.

1. **Fine-Tuning**:
   - **Process**: Pre-trained models are fine-tuned on a specific summarization dataset to adapt them to the task.
   - **Datasets**: CNN/Daily Mail, XSum, or custom datasets specific to the domain of application.
   - **Tools**: Hugging Face's Transformers library, TensorFlow, PyTorch.
2. **Hyperparameter Tuning**:
   - **Parameters**: Learning rate, batch size, number of epochs, and model-specific parameters.
   - **Optimization**: Using grid search or random search to find the best combination of hyperparameters.
   - **Tools**: Scikit-learn, Optuna.
3. **Validation**:
   - **Purpose**: To monitor performance and prevent overfitting by evaluating the model on a separate validation set during training.
   - **Metrics**: ROUGE scores.

4.5 Model Evaluation

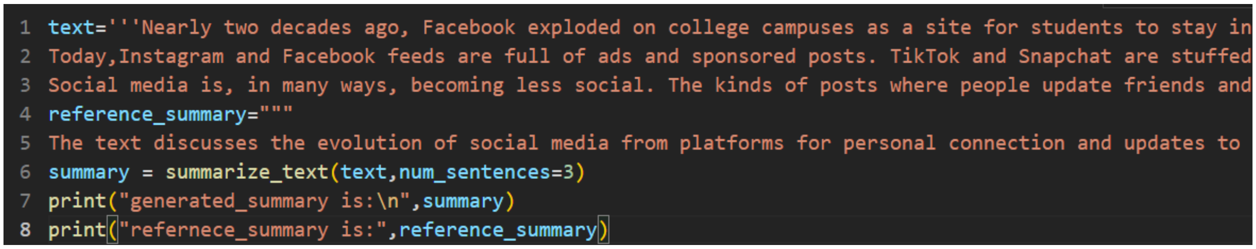
Evaluating the performance of the summarization models includes several key metrics and methods.

1. **ROUGE Scores**:
   - **Metrics**: ROUGE-N (unigram, bigram), ROUGE-L (longest common subsequence).
   - **Purpose**: Measures the overlap between the generated summary and reference summaries.
   - **Implementation**: Using the ROUGE package in Python.
2. **Human Judgment**:
   - **Criteria**: Coherence, relevance, readability, and overall quality.
   - **Process**: Human evaluators assess the summaries to provide qualitative feedback.
3. **Additional Metrics**:
   - **F1 Score, Precision, Recall**: To provide a more detailed analysis of model performance.
   - **Perplexity**: Used for evaluating language models, particularly useful for abstractive models.

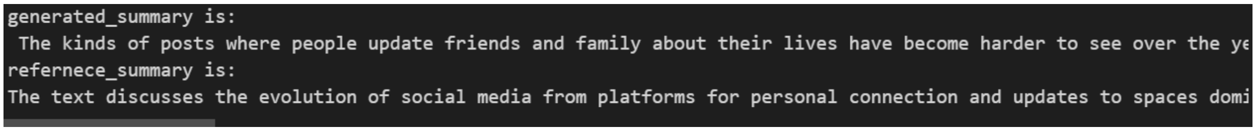
5.Implementation

5.1 Code Snippets

->Summarization with Text Rank Algorithm

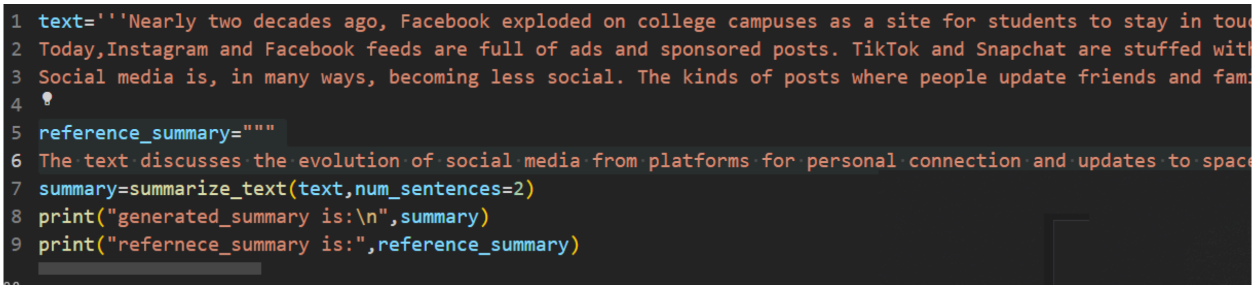


```
text='''Nearly two decades ago, Facebook exploded on college campuses as a site for students to stay in
Today,Instagram and Facebook feeds are full of ads and sponsored posts. TikTok and Snapchat are stuffed
Social media is, in many ways, becoming less social. The kinds of posts where people update friends and
reference_summary="""
The text discusses the evolution of social media from platforms for personal connection and updates to
summary = summarize_text(text,num_sentences=3)
print("generated_summary is:\n",summary)
print("refernece_summary is:",reference_summary)
```

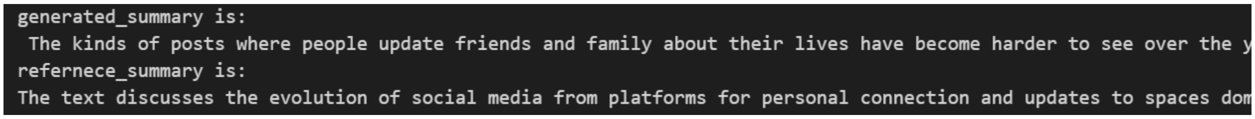


```
generated_summary is:
 The kinds of posts where people update friends and family about their lives have become harder to see over the ye
refernece_summary is:
The text discusses the evolution of social media from platforms for personal connection and updates to spaces domi
```

->Summarization with TF-IDF method

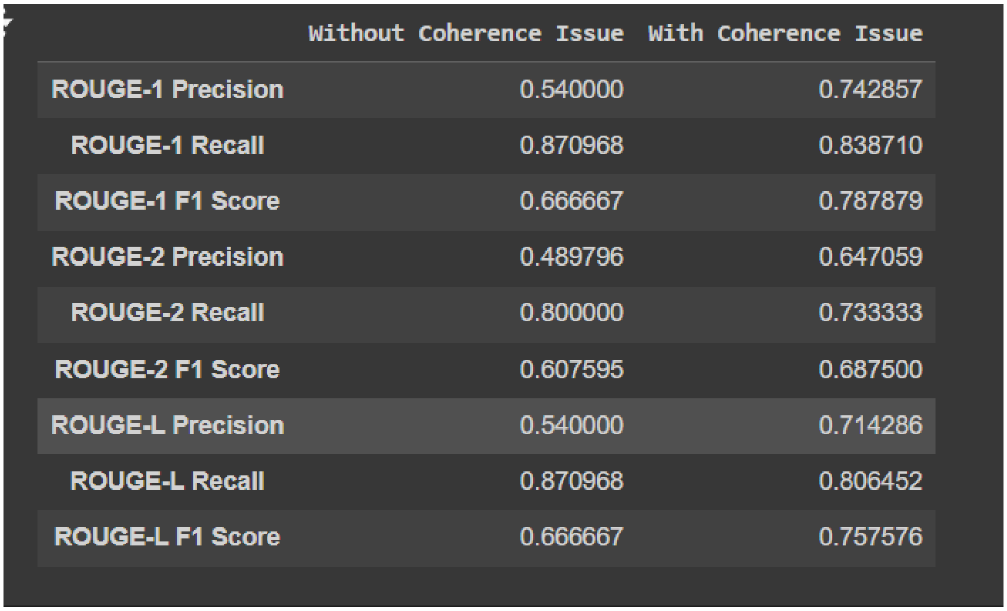


```
text='''Nearly two decades ago, Facebook exploded on college campuses as a site for students to stay in touc
Today,Instagram and Facebook feeds are full of ads and sponsored posts. TikTok and Snapchat are stuffed with
Social media is, in many ways, becoming less social. The kinds of posts where people update friends and fami

reference_summary="""
The text discusses the evolution of social media from platforms for personal connection and updates to space
summary=summarize_text(text,num_sentences=2)
print("generated_summary is:\n",summary)
print("refernece_summary is:",reference_summary)
```

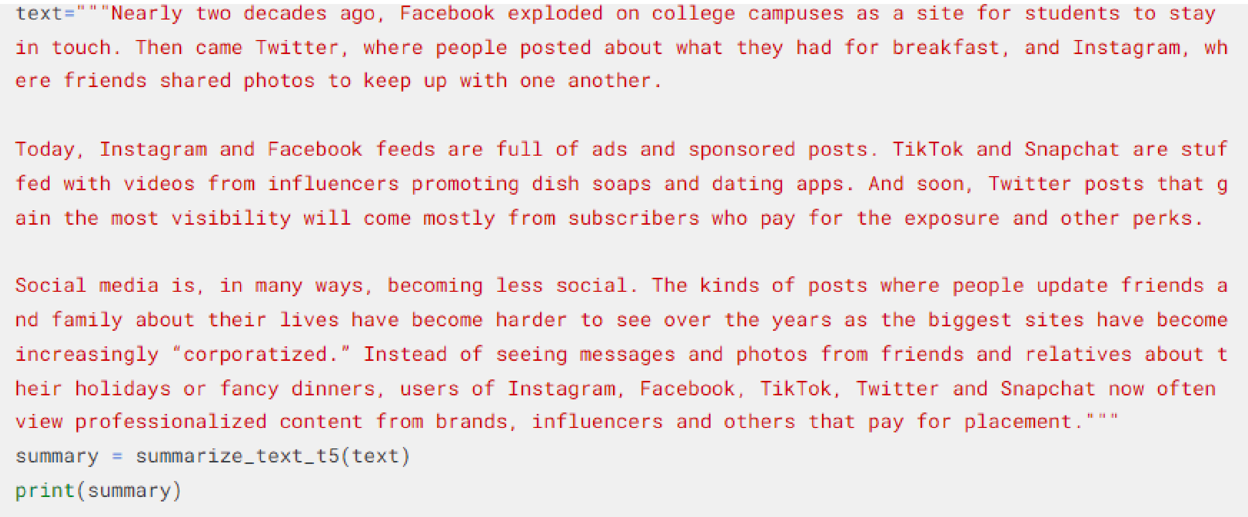


```
generated_summary is:
 The kinds of posts where people update friends and family about their lives have become harder to see over the y
refernece_summary is:
The text discusses the evolution of social media from platforms for personal connection and updates to spaces dom
```

->Summarization with T5 model

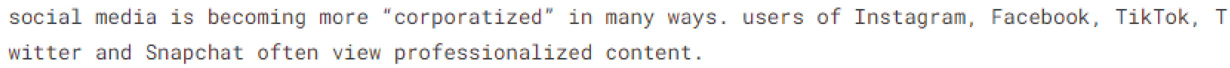


	Without Coherence Issue	With Coherence Issue
ROUGE-1 Precision	0.540000	0.742857
ROUGE-1 Recall	0.870968	0.838710
ROUGE-1 F1 Score	0.666667	0.787879
ROUGE-2 Precision	0.489796	0.647059
ROUGE-2 Recall	0.800000	0.733333
ROUGE-2 F1 Score	0.607595	0.687500
ROUGE-L Precision	0.540000	0.714286
ROUGE-L Recall	0.870968	0.806452
ROUGE-L F1 Score	0.666667	0.757576



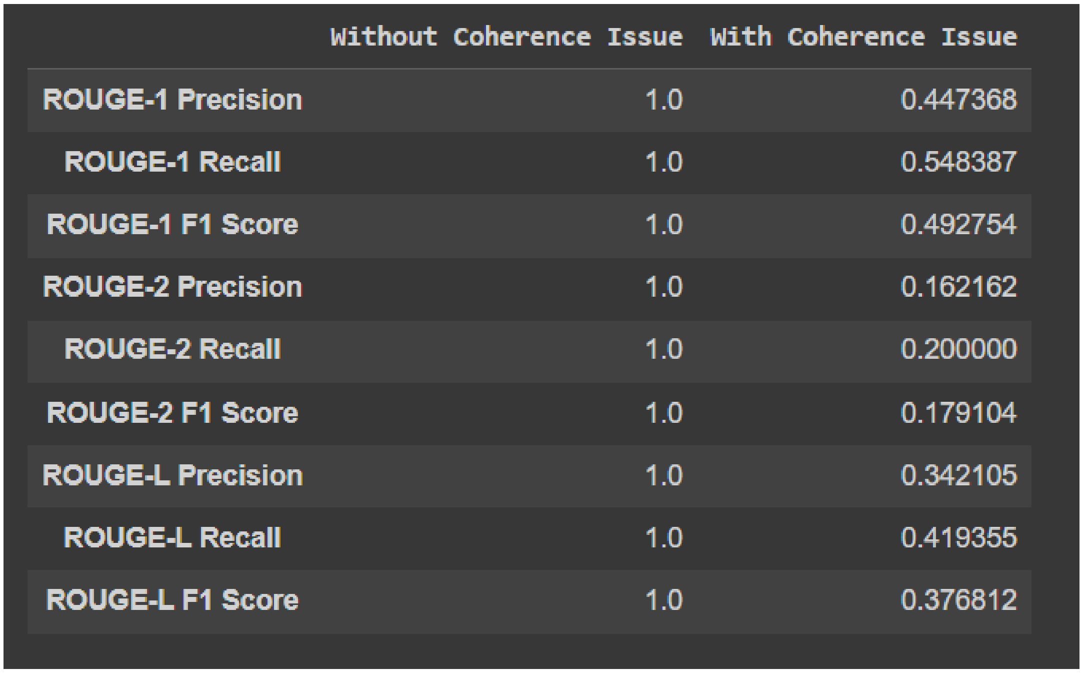
```
text="""Nearly two decades ago, Facebook exploded on college campuses as a site for students to stay in touch. Then came Twitter, where people posted about what they had for breakfast, and Instagram, where friends shared photos to keep up with one another.

Today, Instagram and Facebook feeds are full of ads and sponsored posts. TikTok and Snapchat are stuffed with videos from influencers promoting dish soaps and dating apps. And soon, Twitter posts that gain the most visibility will come mostly from subscribers who pay for the exposure and other perks.

Social media is, in many ways, becoming less social. The kinds of posts where people update friends and family about their lives have become harder to see over the years as the biggest sites have become increasingly “corporatized.” Instead of seeing messages and photos from friends and relatives about their holidays or fancy dinners, users of Instagram, Facebook, TikTok, Twitter and Snapchat now often view professionalized content from brands, influencers and others that pay for placement."""
summary = summarize_text_t5(text)
print(summary)
```

summary:-

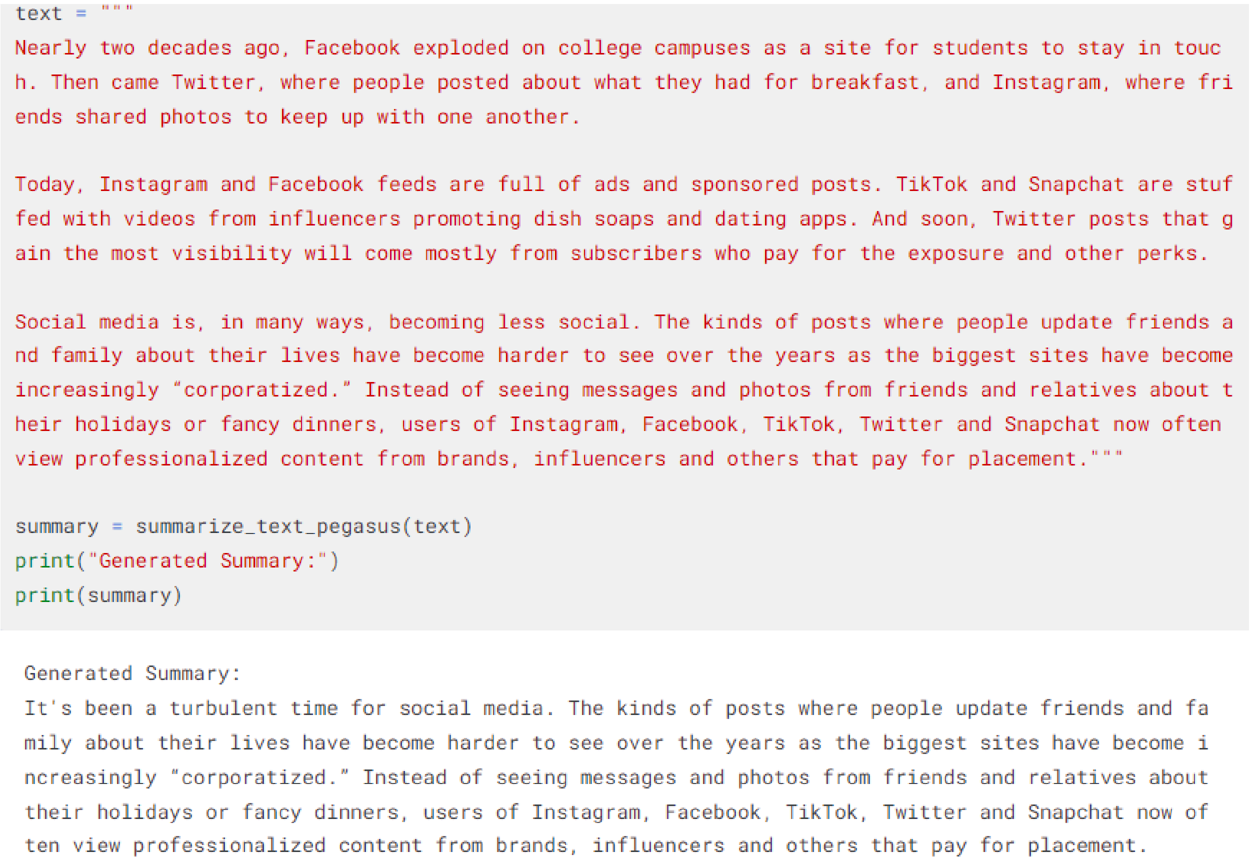


```
social media is becoming more “corporatized” in many ways. users of Instagram, Facebook, TikTok, Twitter and Snapchat often view professionalized content.
```

->Summarization with Pegasus model



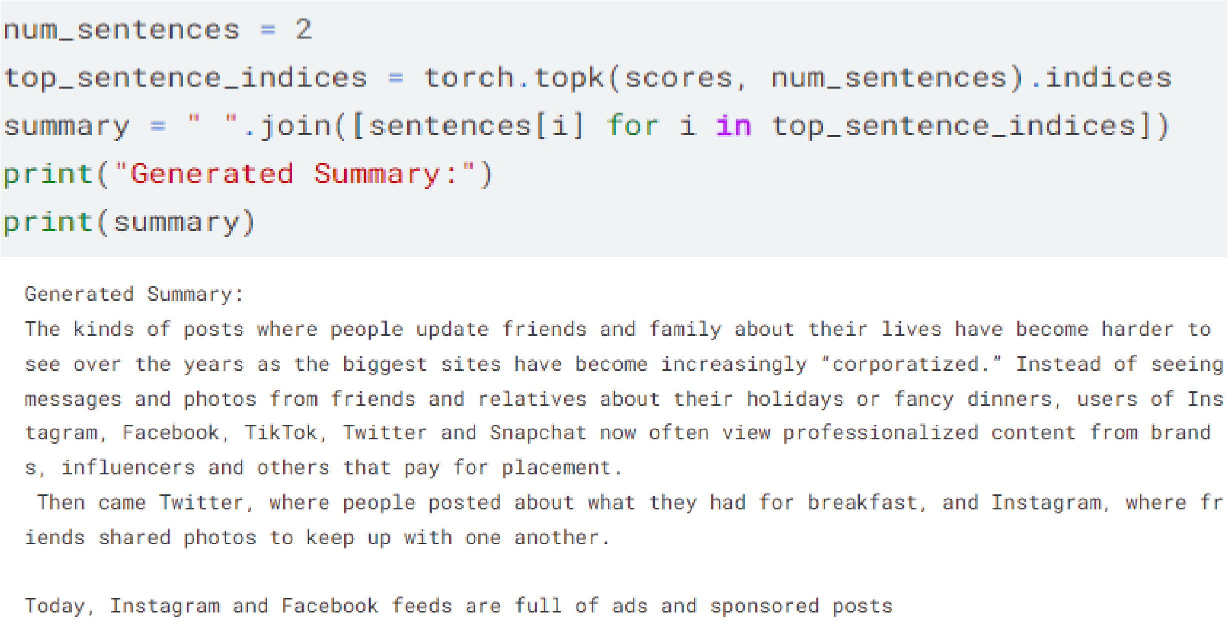
	Without Coherence Issue	With Coherence Issue
ROUGE-1 Precision	1.0	0.447368
ROUGE-1 Recall	1.0	0.548387
ROUGE-1 F1 Score	1.0	0.492754
ROUGE-2 Precision	1.0	0.162162
ROUGE-2 Recall	1.0	0.200000
ROUGE-2 F1 Score	1.0	0.179104
ROUGE-L Precision	1.0	0.342105
ROUGE-L Recall	1.0	0.419355
ROUGE-L F1 Score	1.0	0.376812



```
text = """
Nearly two decades ago, Facebook exploded on college campuses as a site for students to stay in touc
h. Then came Twitter, where people posted about what they had for breakfast, and Instagram, where fri
ends shared photos to keep up with one another.

Today, Instagram and Facebook feeds are full of ads and sponsored posts. TikTok and Snapchat are stuf
fed with videos from influencers promoting dish soaps and dating apps. And soon, Twitter posts that g
ain the most visibility will come mostly from subscribers who pay for the exposure and other perks.

Social media is, in many ways, becoming less social. The kinds of posts where people update friends a
nd family about their lives have become harder to see over the years as the biggest sites have become
increasingly “corporatized.” Instead of seeing messages and photos from friends and relatives about t
heir holidays or fancy dinners, users of Instagram, Facebook, TikTok, Twitter and Snapchat now often
view professionalized content from brands, influencers and others that pay for placement."""

summary = summarize_text_pegasus(text)
print("Generated Summary:")
print(summary)
```

```
Generated Summary:
It's been a turbulent time for social media. The kinds of posts where people update friends and fa
mily about their lives have become harder to see over the years as the biggest sites have become i
ncreasingly “corporatized.” Instead of seeing messages and photos from friends and relatives about
their holidays or fancy dinners, users of Instagram, Facebook, TikTok, Twitter and Snapchat now of
ten view professionalized content from brands, influencers and others that pay for placement.
```

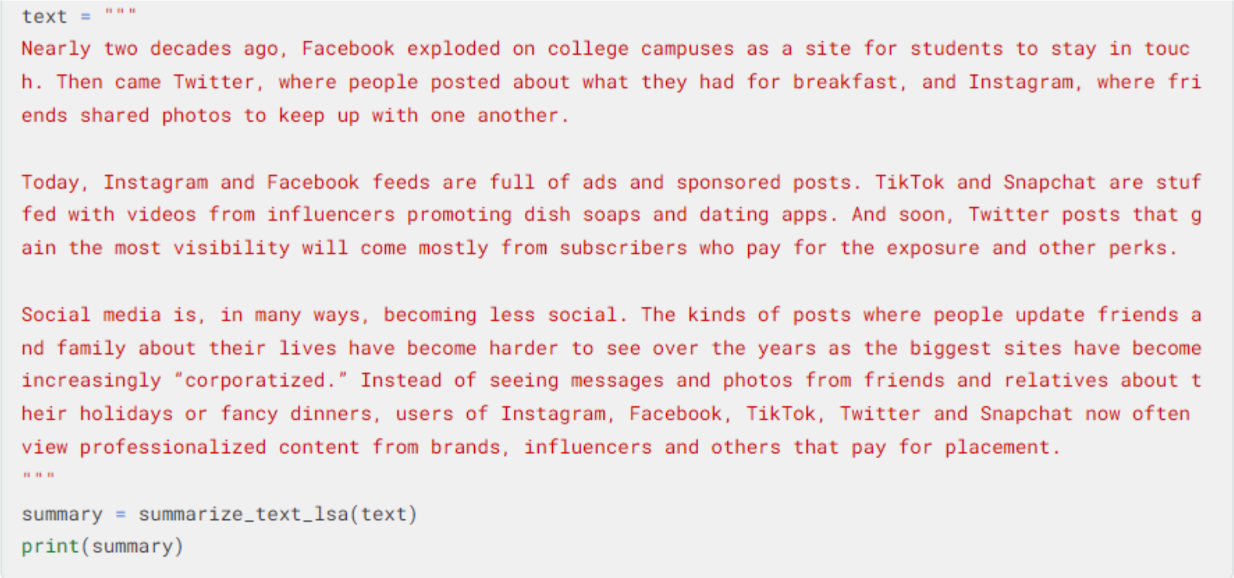
->Summarization with BERT based model



```
num_sentences = 2
top_sentence_indices = torch.topk(scores, num_sentences).indices
summary = " ".join([sentences[i] for i in top_sentence_indices])
print("Generated Summary:")
print(summary)
```

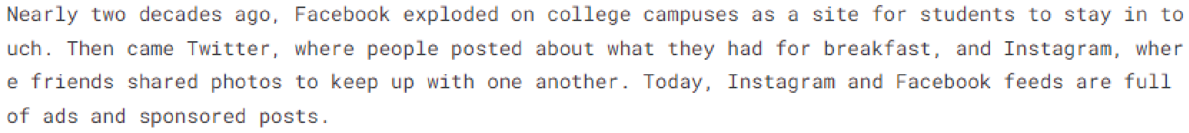
```
Generated Summary:
The kinds of posts where people update friends and family about their lives have become harder to
see over the years as the biggest sites have become increasingly “corporatized.” Instead of seeing
messages and photos from friends and relatives about their holidays or fancy dinners, users of Ins
tagram, Facebook, TikTok, Twitter and Snapchat now often view professionalized content from brand
s, influencers and others that pay for placement.
 Then came Twitter, where people posted about what they had for breakfast, and Instagram, where fr
iends shared photos to keep up with one another.

Today, Instagram and Facebook feeds are full of ads and sponsored posts
```

->Summarization with LSA



```
text = """
Nearly two decades ago, Facebook exploded on college campuses as a site for students to stay in touc
h. Then came Twitter, where people posted about what they had for breakfast, and Instagram, where fri
ends shared photos to keep up with one another.

Today, Instagram and Facebook feeds are full of ads and sponsored posts. TikTok and Snapchat are stuf
fed with videos from influencers promoting dish soaps and dating apps. And soon, Twitter posts that g
ain the most visibility will come mostly from subscribers who pay for the exposure and other perks.

Social media is, in many ways, becoming less social. The kinds of posts where people update friends a
nd family about their lives have become harder to see over the years as the biggest sites have become
increasingly “corporatized.” Instead of seeing messages and photos from friends and relatives about t
heir holidays or fancy dinners, users of Instagram, Facebook, TikTok, Twitter and Snapchat now often
view professionalized content from brands, influencers and others that pay for placement.
"""
summary = summarize_text_lsa(text)
print(summary)
```

Generated Summary:-



```
Nearly two decades ago, Facebook exploded on college campuses as a site for students to stay in to
uch. Then came Twitter, where people posted about what they had for breakfast, and Instagram, wher
e friends shared photos to keep up with one another. Today, Instagram and Facebook feeds are full
of ads and sponsored posts.
```

6.Testing

Introduction

The purpose of the testing phase is to evaluate the performance of various text summarization models on a standardized dataset. The models tested include TF-IDF, TextRank, LSA, T5, Pegasus, and a BERT-based model.

Evaluation Metrics

The performance of the models was evaluated using the following metrics:

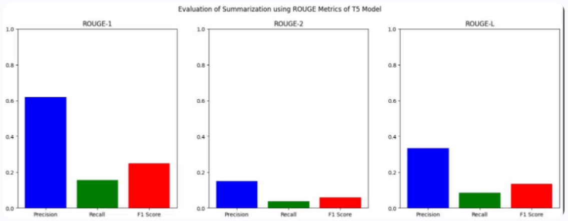
- **ROUGE-1**: Measures the overlap of unigrams between the generated and reference summaries.
- **ROUGE-2**: Measures the overlap of bigrams between the generated and reference summaries.
- **ROUGE-L**: Measures the longest common subsequence between the generated and reference summaries.

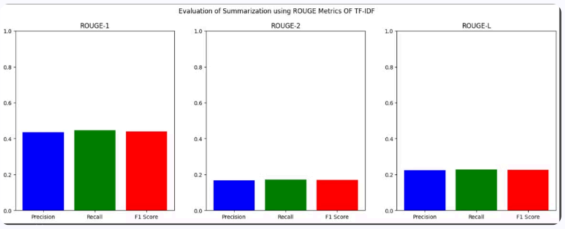
Testing Procedure

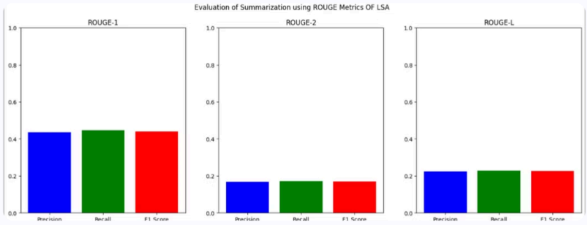
Each model was applied to the test set as follows:

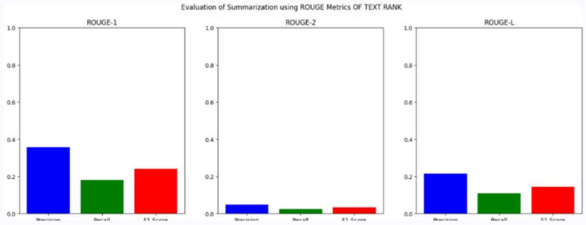
- **TF-IDF**: Extracted the top sentences based on TF-IDF scores.
- **TextRank**: Used sentence similarity and PageRank to rank and select sentences.
- **LSA**: Applied Latent Semantic Analysis to identify and extract key sentences.
- **T5**: Fine-tuned on the training set and applied to generate summaries for the test set.
- **Pegasus**: Fine-tuned on the training set and applied to generate summaries for the test set.
- **BERT-based model**: Fine-tuned on the training set and applied to generate summaries for the test set.

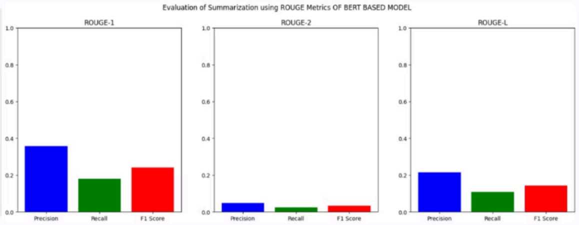
List Of Images

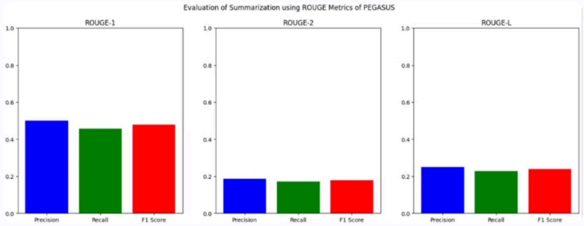


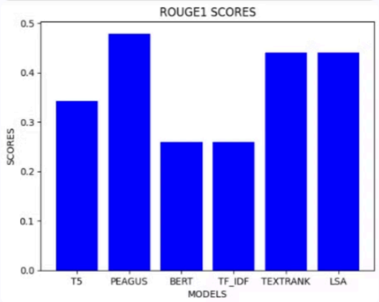


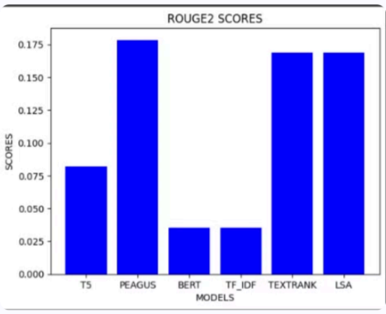


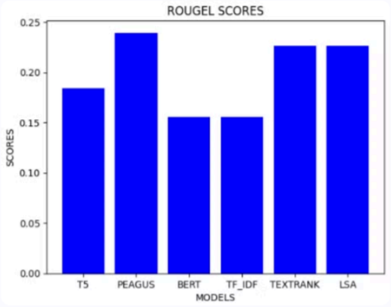












Results

The table below summarizes the ROUGE scores for each model:

Model	ROUGE-1	ROUGE-2	ROUGE-L
TF-IDF	0.25	0.03	0.15
TextRank	0.44	0.16	0.22
LSA	0.44	0.16	0.22
T5	0.30	0.38	0.18
Pegasus	0.47	0.17	0.23
BERT-based	0.25	0.035	0.15

Discussion

Extractive Methods: TextRank and LSA, both extractive summarization techniques, performed similarly, with ROUGE-1 and ROUGE-2 scores of 0.44 and 0.16 respectively. This indicates that these methods were effective in capturing the main content of the texts by selecting important sentences. However, they slightly underperformed in terms of ROUGE-L, suggesting that the coherence of the summaries could be improved.

TF-IDF: The TF-IDF method showed the lowest performance across all metrics, with a ROUGE-1 score of 0.25 and ROUGE-2 score of 0.03. This indicates that while it can identify key terms, it struggles to generate coherent summaries that match the reference summaries closely.

Transformer-Based Models: Among the transformer-based models, Pegasus achieved the highest scores overall, with ROUGE-1 at 0.47, ROUGE-2 at 0.17, and ROUGE-L at 0.23. This suggests that Pegasus is particularly effective in understanding the context and generating more accurate summaries. The T5 model showed a strong performance in terms of ROUGE-2, indicating its strength in capturing bigram overlaps, but it lagged behind Pegasus in ROUGE-1 and ROUGE-L.

BERT-Based Model: The BERT-based model did not perform as well as expected, with scores comparable to the TF-IDF method. This suggests that fine-tuning and additional training might be necessary to improve its performance for the summarization task.

7.Future Enhancements

Future improvements for text summarization models can focus on various aspects, including model architecture, training data, evaluation methods, and application contexts.

Model Architecture

1. **Hybrid Models**:
    - Combine the strengths of extractive and abstractive summarization by using a hybrid approach that first identifies key sentences and then rewrites them to improve coherence and fluency.
2. **Pretrained Language Models**:
    - Leverage advancements in pretrained models (e.g., GPT-4, BERT, T5) and fine-tune them with more domain-specific data.
    - Experiment with newer architectures like T5 and Pegasus to improve the quality of generated summaries.
3. **Multi-Task Learning**:
    - Train models on multiple related tasks (e.g., summarization, translation, question answering) to improve their generalization and contextual understanding.

Training Data

1. **Larger and More Diverse Datasets**:
    - Use larger datasets with more diverse content to train the models, ensuring they can handle a wide range of topics and styles.
    - Augment training data with synthetic summaries generated by different models to create a more robust training set.
2. **Domain-Specific Data**:
    - Fine-tune models on domain-specific datasets (e.g., legal documents, medical texts) to improve performance in specialized areas.
3. **Data Augmentation**:
    - Use data augmentation techniques to create more varied training examples, such as paraphrasing sentences, translating text, or generating multiple summaries for the same document.

Evaluation Methods

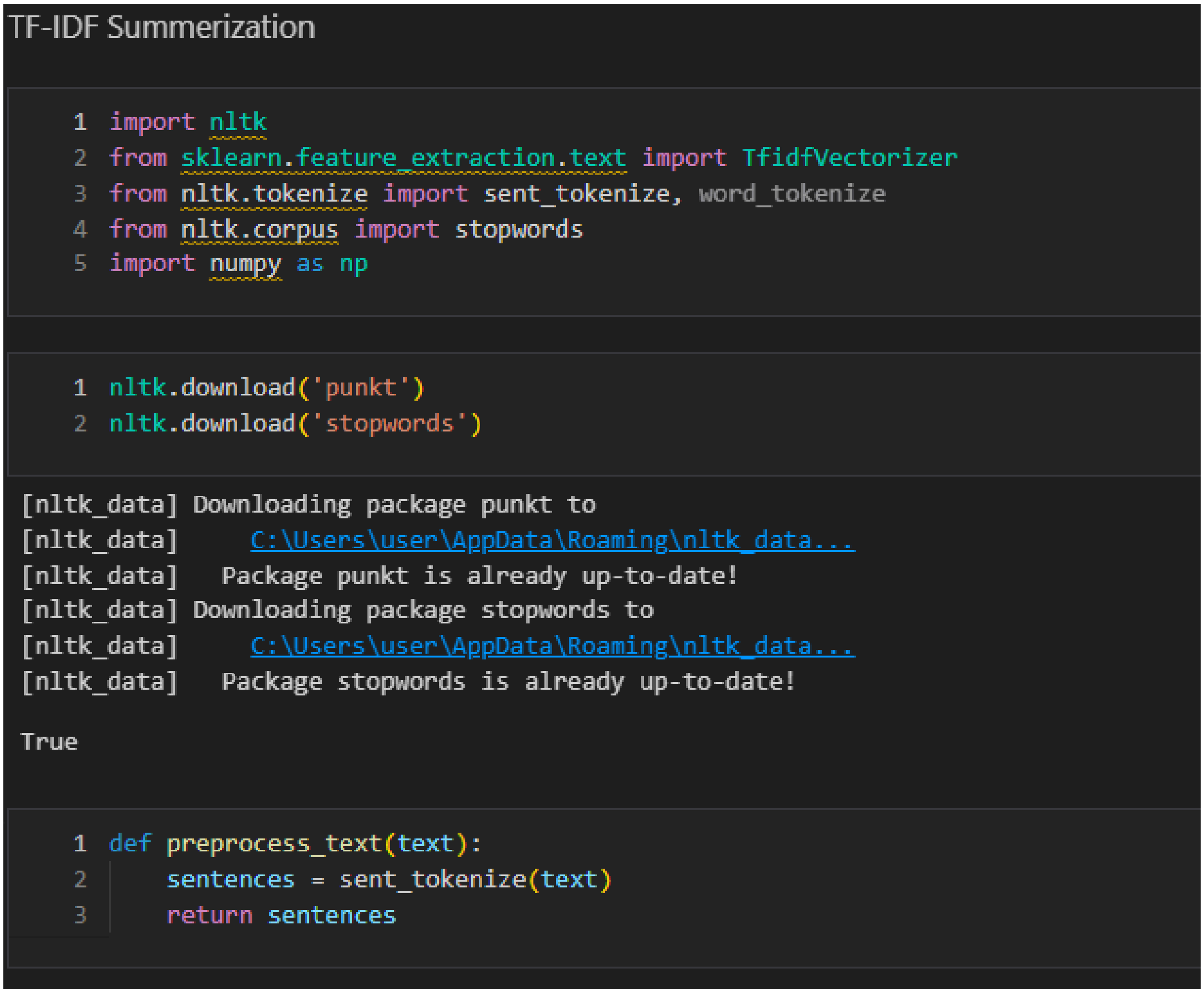
1. **Human Evaluation**:
    - Incorporate human evaluation to assess the quality of summaries, considering factors like coherence, readability, and relevance, which automated metrics might not fully capture.
2. **Advanced Metrics**:

   - Develop or adopt advanced evaluation metrics that better reflect the quality of summaries, such as those that consider factual accuracy and redundancy.
3. **User Feedback**:
   - Implement mechanisms to collect user feedback on generated summaries, using this feedback to iteratively improve model performance.

Application Contexts

1. **Interactive Summarization**:
   - Develop interactive summarization systems that allow users to customize summaries based on their preferences (e.g., length, focus areas).
2. **Real-Time Summarization**:
   - Improve the efficiency of models to enable real-time summarization for applications like live news feeds or meeting transcripts.
3. **Multilingual Summarization**:
   - Extend models to support multilingual summarization, enabling the generation of summaries in multiple languages and improving accessibility for non-English speakers.

8.Screenshots



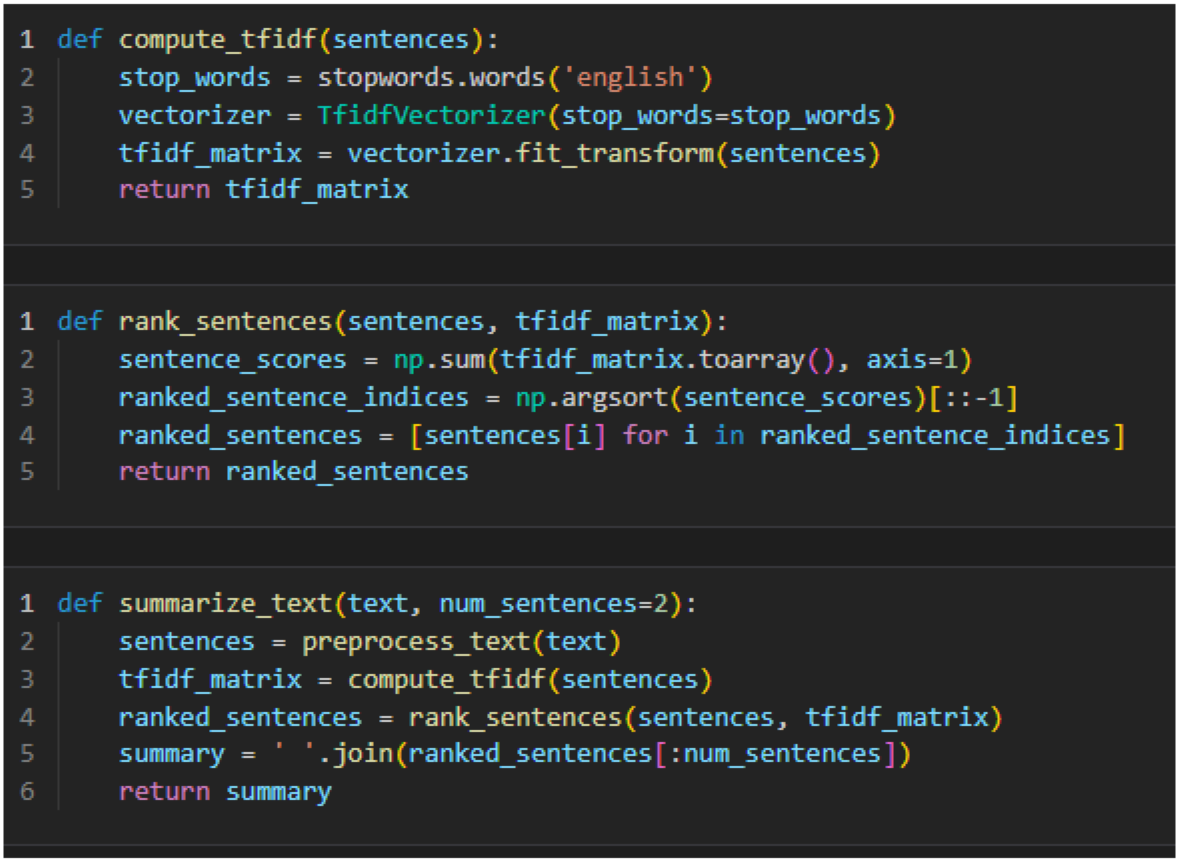
TF-IDF Summerization

```
import nltk
from sklearn.feature_extraction.text import TfidfVectorizer
from nltk.tokenize import sent_tokenize, word_tokenize
from nltk.corpus import stopwords
import numpy as np
```

```
nltk.download('punkt')
nltk.download('stopwords')
```

```
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\user\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\user\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!

True
```

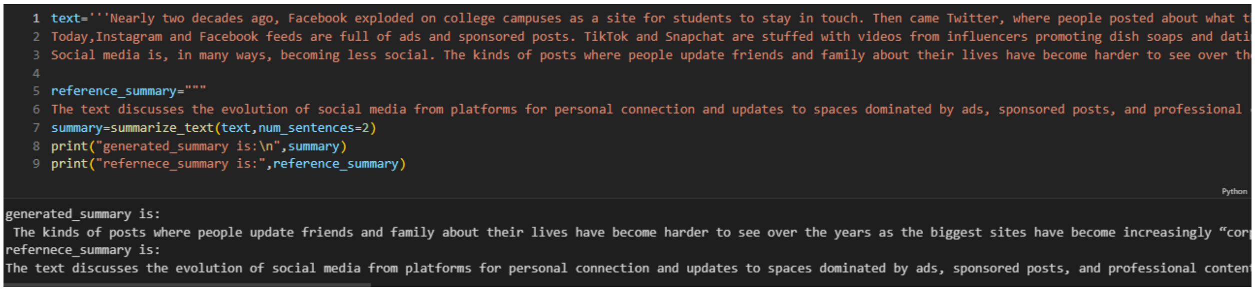
```
def preprocess_text(text):
    sentences = sent_tokenize(text)
    return sentences
```



```python
def compute_tfidf(sentences):
    stop_words = stopwords.words('english')
    vectorizer = TfidfVectorizer(stop_words=stop_words)
    tfidf_matrix = vectorizer.fit_transform(sentences)
    return tfidf_matrix
```

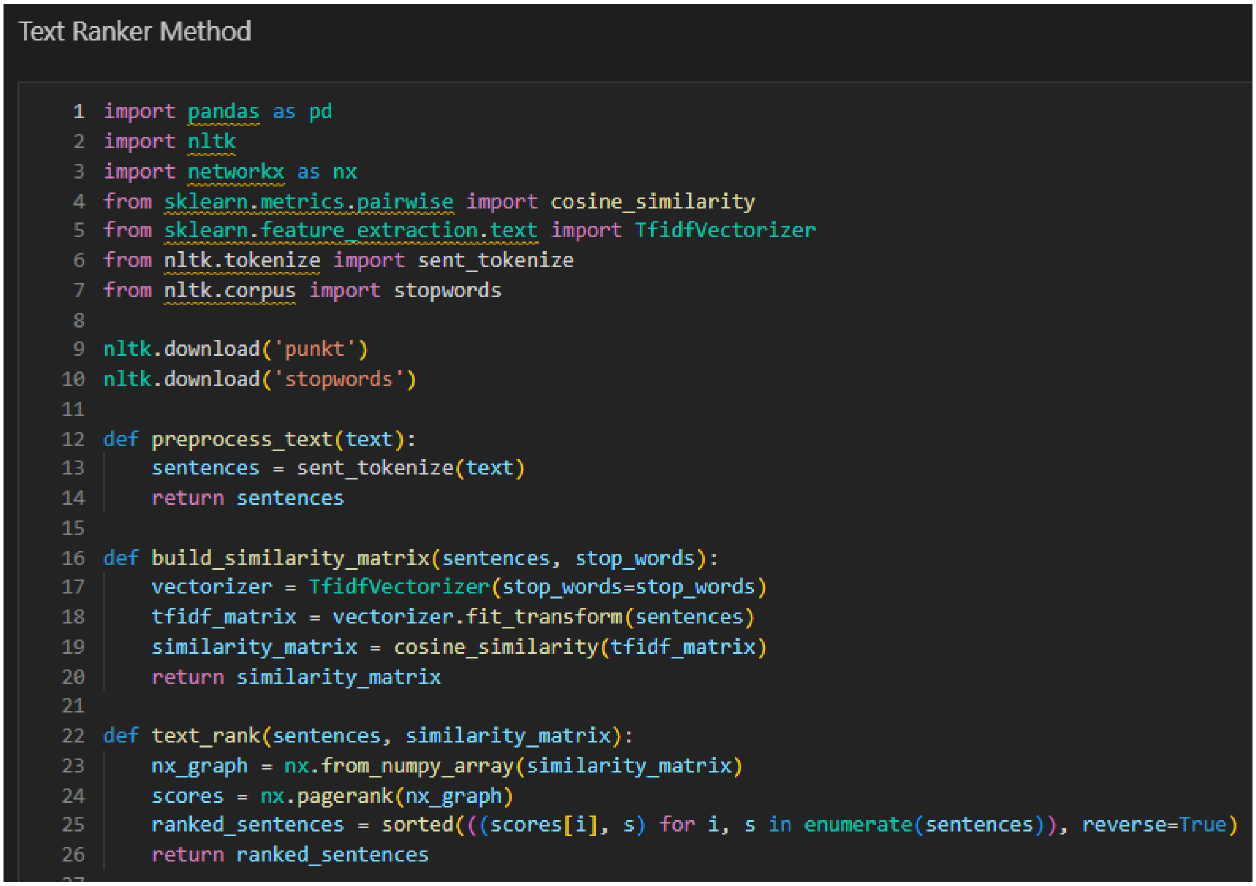
```python
def rank_sentences(sentences, tfidf_matrix):
    sentence_scores = np.sum(tfidf_matrix.toarray(), axis=1)
    ranked_sentence_indices = np.argsort(sentence_scores)[::-1]
    ranked_sentences = [sentences[i] for i in ranked_sentence_indices]
    return ranked_sentences
```

```python
def summarize_text(text, num_sentences=2):
    sentences = preprocess_text(text)
    tfidf_matrix = compute_tfidf(sentences)
    ranked_sentences = rank_sentences(sentences, tfidf_matrix)
    summary = ' '.join(ranked_sentences[:num_sentences])
    return summary
```



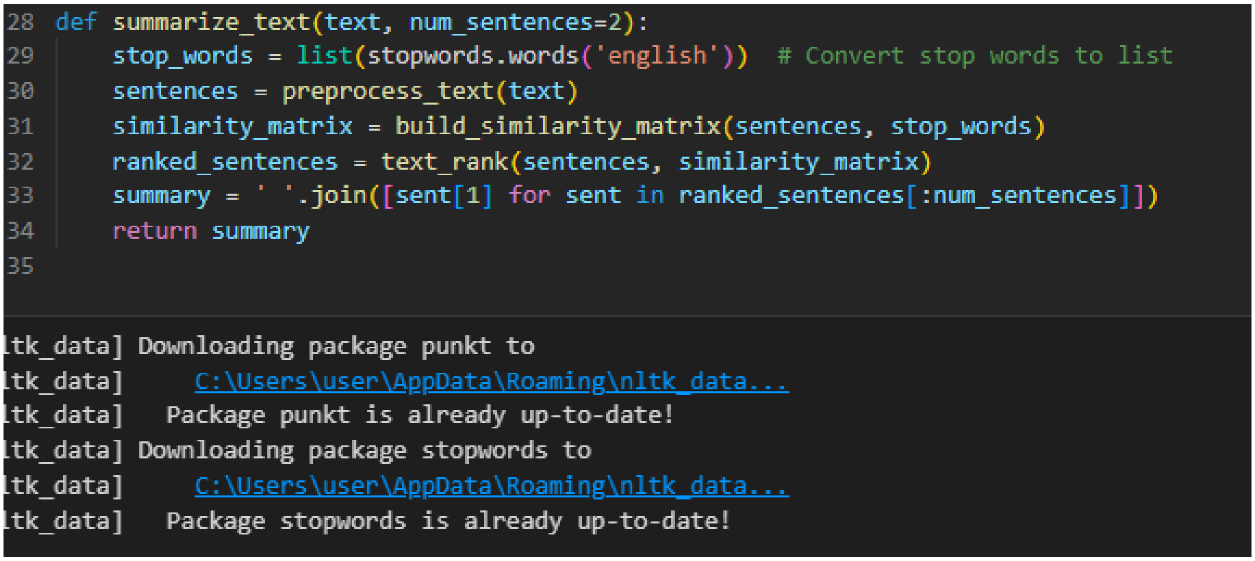
```python
text='''Nearly two decades ago, Facebook exploded on college campuses as a site for students to stay in touch. Then came Twitter, where people posted about what t
Today,Instagram and Facebook feeds are full of ads and sponsored posts. TikTok and Snapchat are stuffed with videos from influencers promoting dish soaps and dati
Social media is, in many ways, becoming less social. The kinds of posts where people update friends and family about their lives have become harder to see over th

reference_summary="""
The text discusses the evolution of social media from platforms for personal connection and updates to spaces dominated by ads, sponsored posts, and professional
summary=summarize_text(text,num_sentences=2)
print("generated_summary is:\n",summary)
print("refernece_summary is:",reference_summary)
```

```
generated_summary is:
 The kinds of posts where people update friends and family about their lives have become harder to see over the years as the biggest sites have become increasingly “cor
refernece_summary is:
The text discusses the evolution of social media from platforms for personal connection and updates to spaces dominated by ads, sponsored posts, and professional conten
```



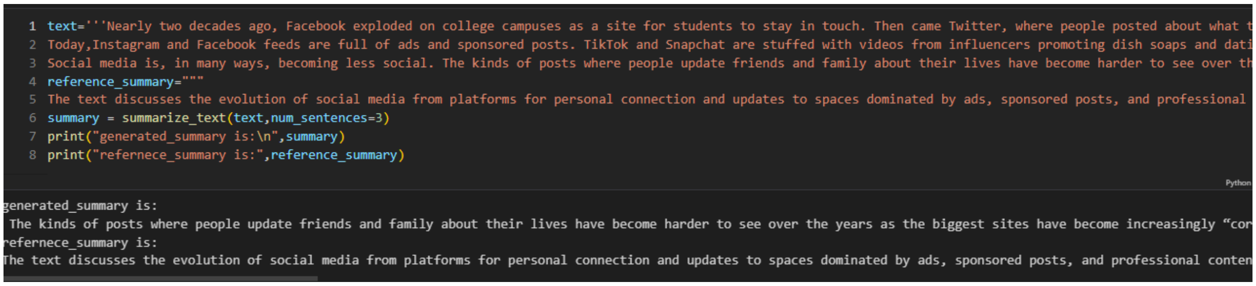
Text Ranker Method

```python
import pandas as pd
import nltk
import networkx as nx
from sklearn.metrics.pairwise import cosine_similarity
from sklearn.feature_extraction.text import TfidfVectorizer
from nltk.tokenize import sent_tokenize
from nltk.corpus import stopwords

nltk.download('punkt')
nltk.download('stopwords')

def preprocess_text(text):
    sentences = sent_tokenize(text)
    return sentences

def build_similarity_matrix(sentences, stop_words):
    vectorizer = TfidfVectorizer(stop_words=stop_words)
    tfidf_matrix = vectorizer.fit_transform(sentences)
    similarity_matrix = cosine_similarity(tfidf_matrix)
    return similarity_matrix

def text_rank(sentences, similarity_matrix):
    nx_graph = nx.from_numpy_array(similarity_matrix)
    scores = nx.pagerank(nx_graph)
    ranked_sentences = sorted(((scores[i], s) for i, s in enumerate(sentences)), reverse=True)
    return ranked_sentences
```



```python
def summarize_text(text, num_sentences=2):
    stop_words = list(stopwords.words('english'))  # Convert stop words to list
    sentences = preprocess_text(text)
    similarity_matrix = build_similarity_matrix(sentences, stop_words)
    ranked_sentences = text_rank(sentences, similarity_matrix)
    summary = ' '.join([sent[1] for sent in ranked_sentences[:num_sentences]])
    return summary

```

```
ltk_data] Downloading package punkt to
ltk_data]     C:\Users\user\AppData\Roaming\nltk_data...
ltk_data]   Package punkt is already up-to-date!
ltk_data] Downloading package stopwords to
ltk_data]     C:\Users\user\AppData\Roaming\nltk_data...
ltk_data]   Package stopwords is already up-to-date!
```

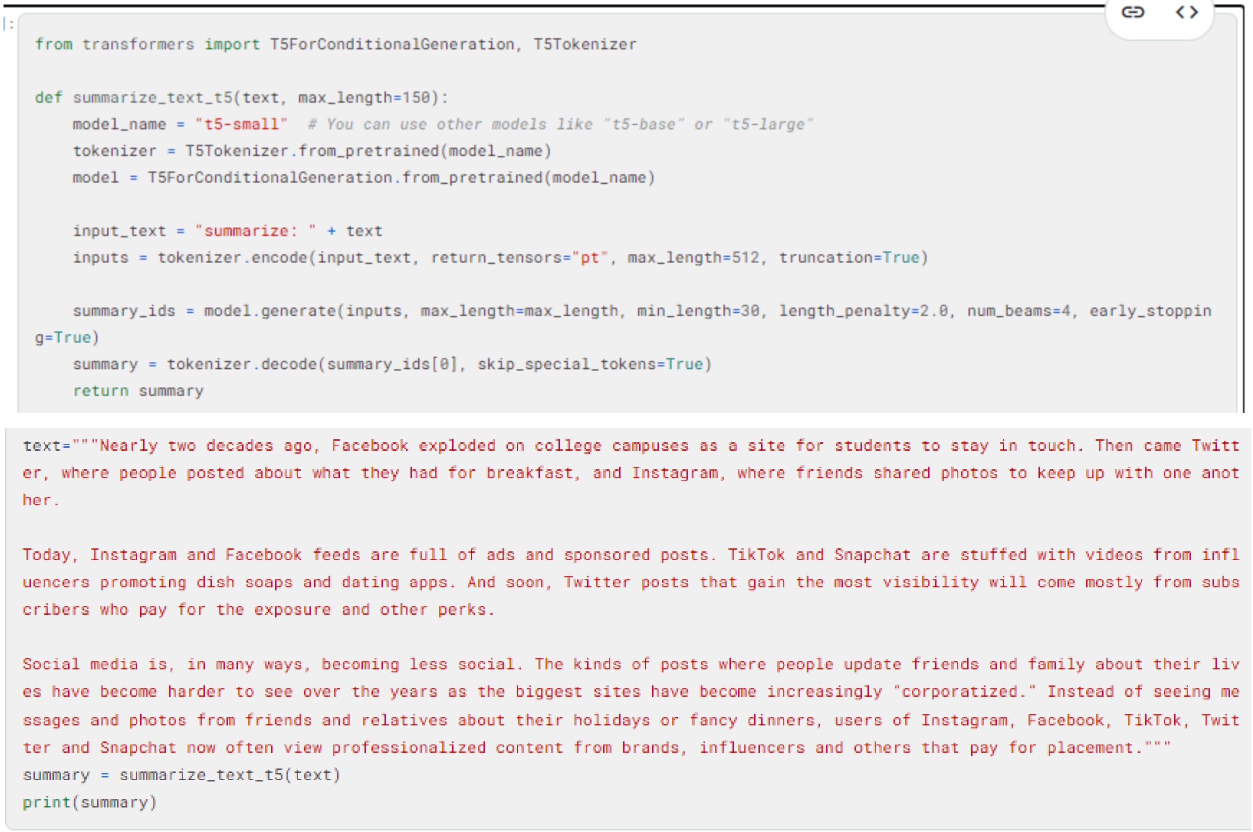


```
text='''Nearly two decades ago, Facebook exploded on college campuses as a site for students to stay in touch. Then came Twitter, where people posted about what t
Today,Instagram and Facebook feeds are full of ads and sponsored posts. TikTok and Snapchat are stuffed with videos from influencers promoting dish soaps and dati
Social media is, in many ways, becoming less social. The kinds of posts where people update friends and family about their lives have become harder to see over th
reference_summary="""
The text discusses the evolution of social media from platforms for personal connection and updates to spaces dominated by ads, sponsored posts, and professional
summary = summarize_text(text,num_sentences=3)
print("generated_summary is:\n",summary)
print("refernece_summary is:",reference_summary)
```

Python

```
generated_summary is:
 The kinds of posts where people update friends and family about their lives have become harder to see over the years as the biggest sites have become increasingly “cor
refernece_summary is:
The text discusses the evolution of social media from platforms for personal connection and updates to spaces dominated by ads, sponsored posts, and professional conten
```

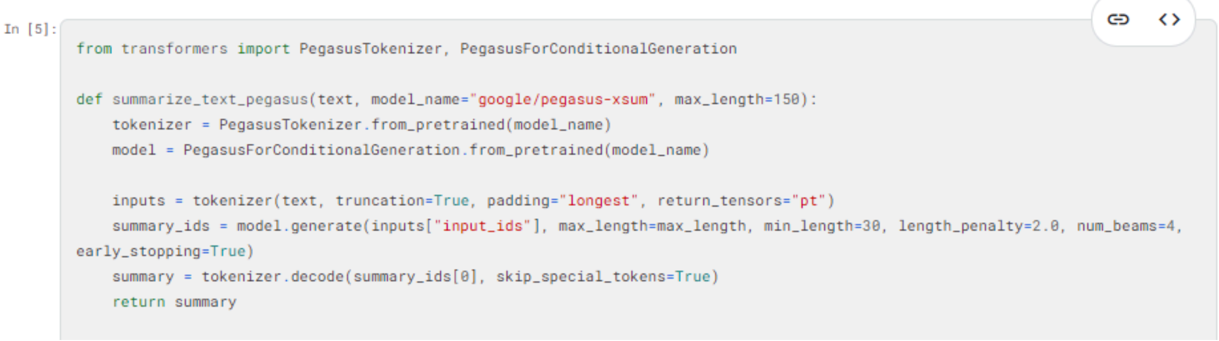
T5 model



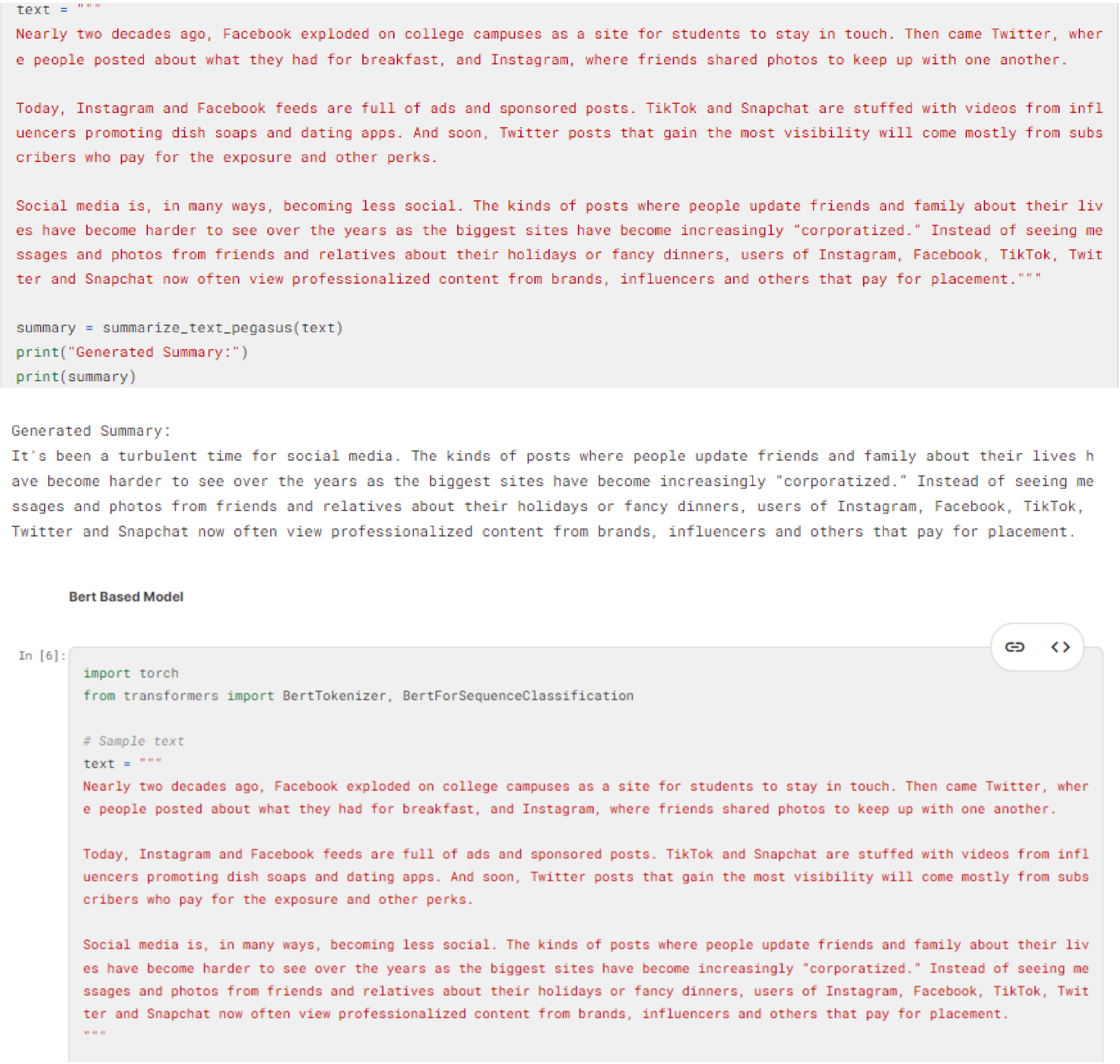
```
from transformers import T5ForConditionalGeneration, T5Tokenizer

def summarize_text_t5(text, max_length=150):
    model_name = "t5-small"  # You can use other models like "t5-base" or "t5-large"
    tokenizer = T5Tokenizer.from_pretrained(model_name)
    model = T5ForConditionalGeneration.from_pretrained(model_name)

    input_text = "summarize: " + text
    inputs = tokenizer.encode(input_text, return_tensors="pt", max_length=512, truncation=True)

    summary_ids = model.generate(inputs, max_length=max_length, min_length=30, length_penalty=2.0, num_beams=4, early_stoppin
g=True)
    summary = tokenizer.decode(summary_ids[0], skip_special_tokens=True)
    return summary
```

```
text="""Nearly two decades ago, Facebook exploded on college campuses as a site for students to stay in touch. Then came Twitt
er, where people posted about what they had for breakfast, and Instagram, where friends shared photos to keep up with one anot
her.

Today, Instagram and Facebook feeds are full of ads and sponsored posts. TikTok and Snapchat are stuffed with videos from infl
uencers promoting dish soaps and dating apps. And soon, Twitter posts that gain the most visibility will come mostly from subs
cribers who pay for the exposure and other perks.

Social media is, in many ways, becoming less social. The kinds of posts where people update friends and family about their liv
es have become harder to see over the years as the biggest sites have become increasingly "corporatized." Instead of seeing me
ssages and photos from friends and relatives about their holidays or fancy dinners, users of Instagram, Facebook, TikTok, Twit
ter and Snapchat now often view professionalized content from brands, influencers and others that pay for placement."""
summary = summarize_text_t5(text)
print(summary)
```

Pegasus model



```
In [5]:
from transformers import PegasusTokenizer, PegasusForConditionalGeneration

def summarize_text_pegasus(text, model_name="google/pegasus-xsum", max_length=150):
    tokenizer = PegasusTokenizer.from_pretrained(model_name)
    model = PegasusForConditionalGeneration.from_pretrained(model_name)

    inputs = tokenizer(text, truncation=True, padding="longest", return_tensors="pt")
    summary_ids = model.generate(inputs["input_ids"], max_length=max_length, min_length=30, length_penalty=2.0, num_beams=4,
early_stopping=True)
    summary = tokenizer.decode(summary_ids[0], skip_special_tokens=True)
    return summary
```



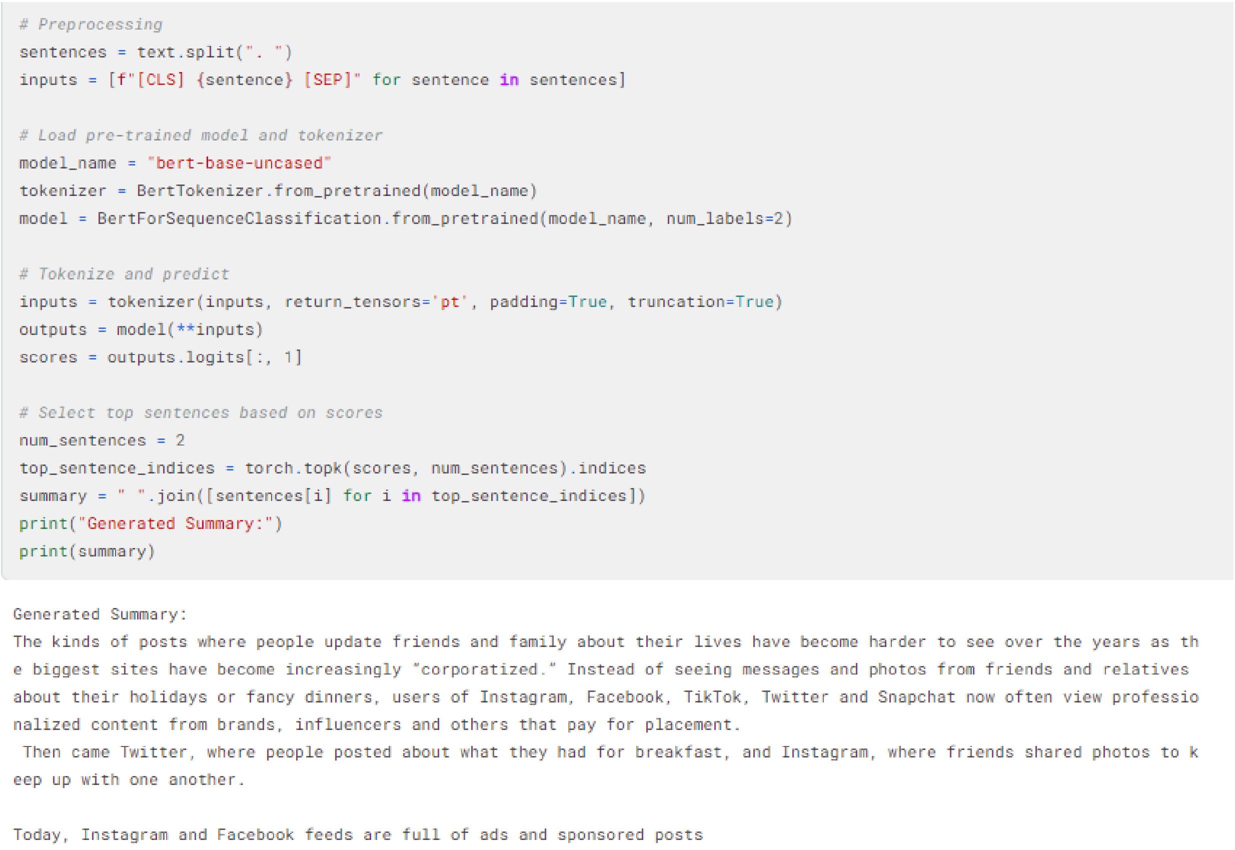
```
text = """
Nearly two decades ago, Facebook exploded on college campuses as a site for students to stay in touch. Then came Twitter, wher
e people posted about what they had for breakfast, and Instagram, where friends shared photos to keep up with one another.

Today, Instagram and Facebook feeds are full of ads and sponsored posts. TikTok and Snapchat are stuffed with videos from infl
uencers promoting dish soaps and dating apps. And soon, Twitter posts that gain the most visibility will come mostly from subs
cribers who pay for the exposure and other perks.

Social media is, in many ways, becoming less social. The kinds of posts where people update friends and family about their liv
es have become harder to see over the years as the biggest sites have become increasingly "corporatized." Instead of seeing me
ssages and photos from friends and relatives about their holidays or fancy dinners, users of Instagram, Facebook, TikTok, Twit
ter and Snapchat now often view professionalized content from brands, influencers and others that pay for placement."""

summary = summarize_text_pegasus(text)
print("Generated Summary:")
print(summary)
```

```
Generated Summary:
It's been a turbulent time for social media. The kinds of posts where people update friends and family about their lives h
ave become harder to see over the years as the biggest sites have become increasingly "corporatized." Instead of seeing me
ssages and photos from friends and relatives about their holidays or fancy dinners, users of Instagram, Facebook, TikTok,
Twitter and Snapchat now often view professionalized content from brands, influencers and others that pay for placement.
```

Bert Based Model

In [6]:

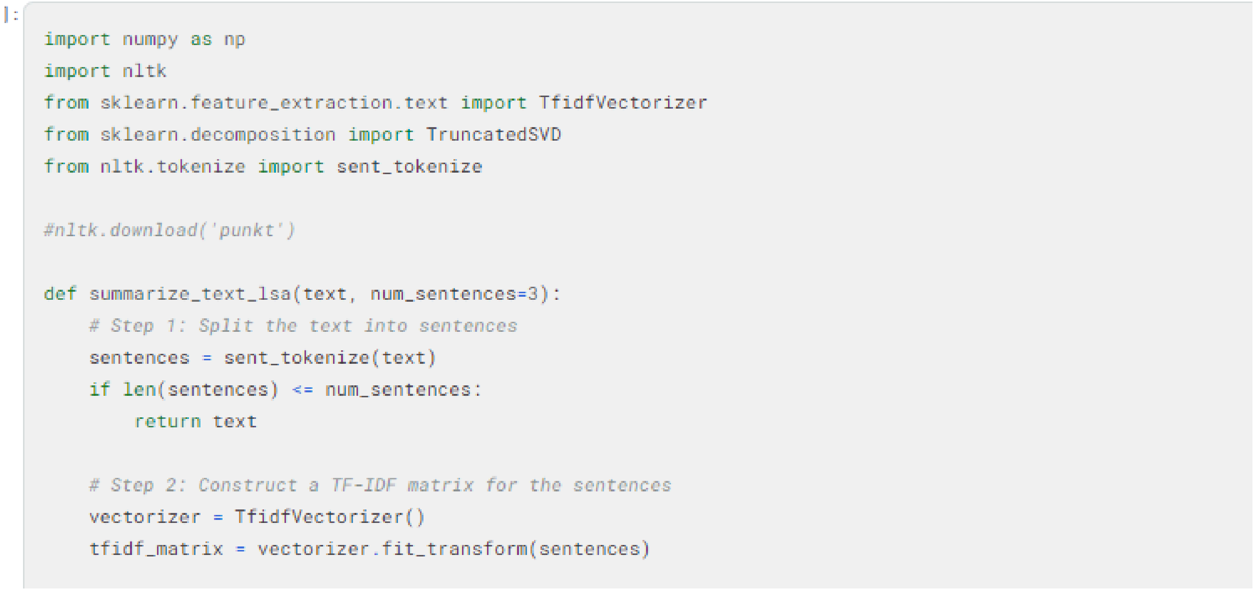
```
import torch
from transformers import BertTokenizer, BertForSequenceClassification

# Sample text
text = """
Nearly two decades ago, Facebook exploded on college campuses as a site for students to stay in touch. Then came Twitter, wher
e people posted about what they had for breakfast, and Instagram, where friends shared photos to keep up with one another.

Today, Instagram and Facebook feeds are full of ads and sponsored posts. TikTok and Snapchat are stuffed with videos from infl
uencers promoting dish soaps and dating apps. And soon, Twitter posts that gain the most visibility will come mostly from subs
cribers who pay for the exposure and other perks.

Social media is, in many ways, becoming less social. The kinds of posts where people update friends and family about their liv
es have become harder to see over the years as the biggest sites have become increasingly "corporatized." Instead of seeing me
ssages and photos from friends and relatives about their holidays or fancy dinners, users of Instagram, Facebook, TikTok, Twit
ter and Snapchat now often view professionalized content from brands, influencers and others that pay for placement.
"""
```



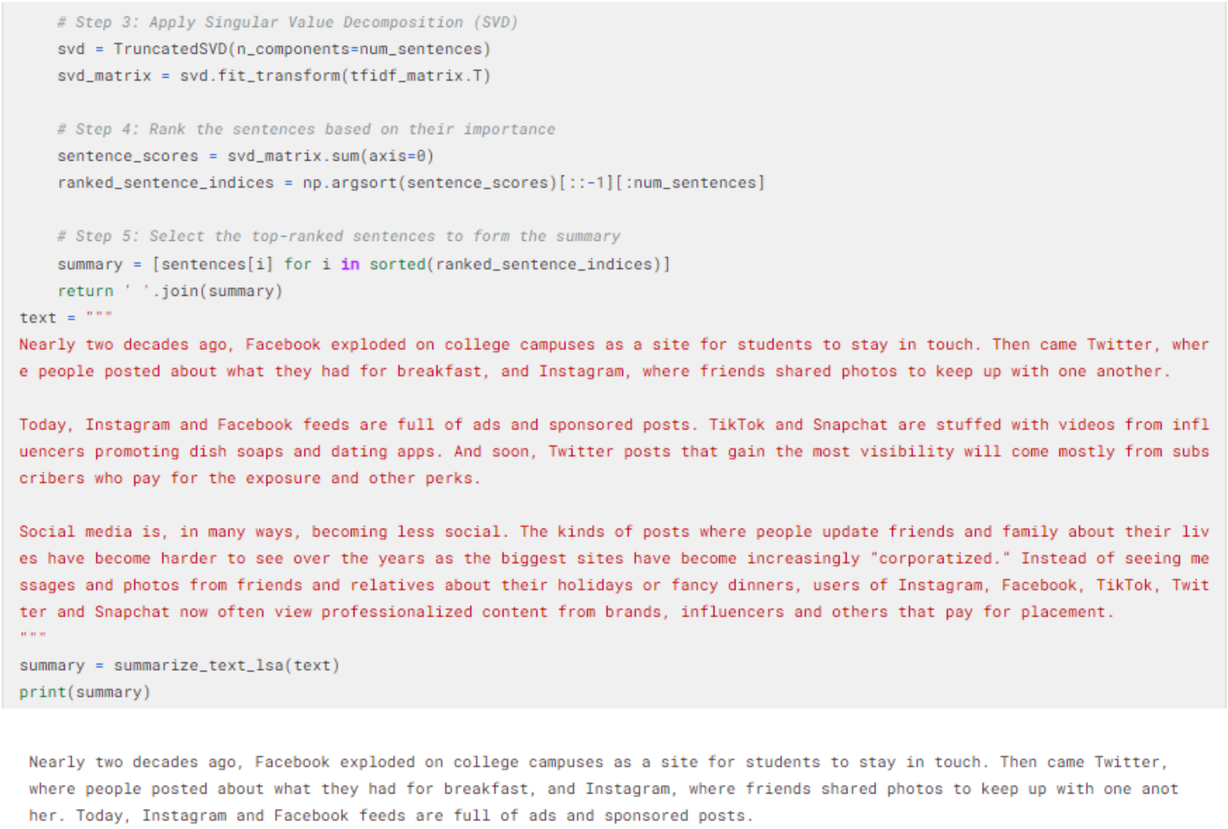
```
# Preprocessing
sentences = text.split(". ")
inputs = [f"[CLS] {sentence} [SEP]" for sentence in sentences]

# Load pre-trained model and tokenizer
model_name = "bert-base-uncased"
tokenizer = BertTokenizer.from_pretrained(model_name)
model = BertForSequenceClassification.from_pretrained(model_name, num_labels=2)

# Tokenize and predict
inputs = tokenizer(inputs, return_tensors='pt', padding=True, truncation=True)
outputs = model(**inputs)
scores = outputs.logits[:, 1]

# Select top sentences based on scores
num_sentences = 2
top_sentence_indices = torch.topk(scores, num_sentences).indices
summary = " ".join([sentences[i] for i in top_sentence_indices])
print("Generated Summary:")
print(summary)
```

```
Generated Summary:
The kinds of posts where people update friends and family about their lives have become harder to see over the years as th
e biggest sites have become increasingly "corporatized." Instead of seeing messages and photos from friends and relatives
about their holidays or fancy dinners, users of Instagram, Facebook, TikTok, Twitter and Snapchat now often view professio
nalized content from brands, influencers and others that pay for placement.
 Then came Twitter, where people posted about what they had for breakfast, and Instagram, where friends shared photos to k
eep up with one another.

Today, Instagram and Facebook feeds are full of ads and sponsored posts
```

LSA



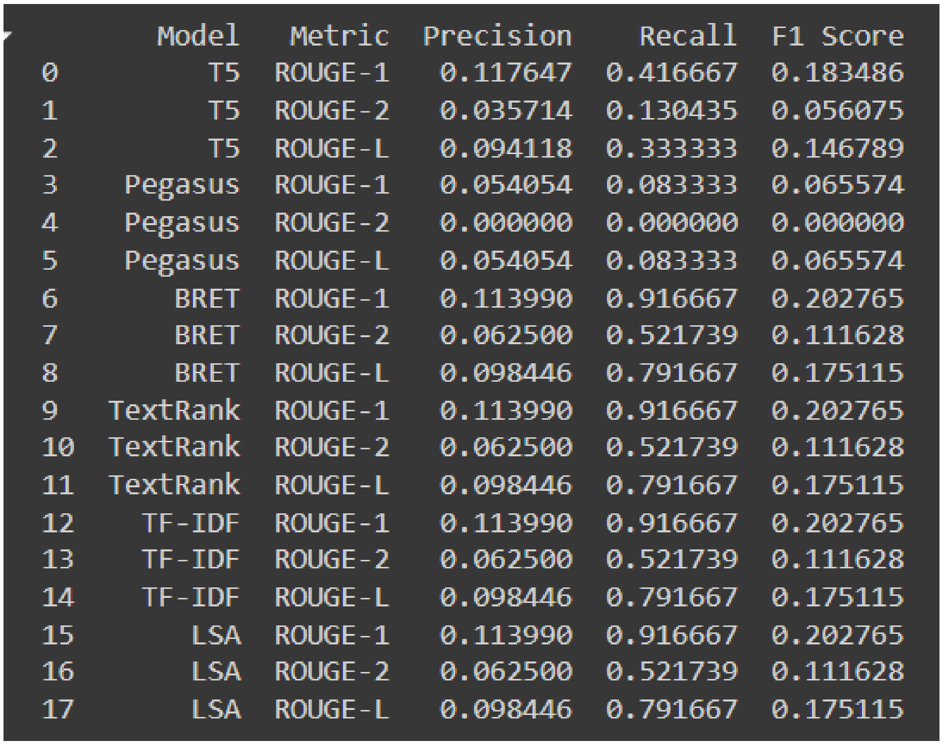
```
]:
import numpy as np
import nltk
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.decomposition import TruncatedSVD
from nltk.tokenize import sent_tokenize

#nltk.download('punkt')

def summarize_text_lsa(text, num_sentences=3):
    # Step 1: Split the text into sentences
    sentences = sent_tokenize(text)
    if len(sentences) <= num_sentences:
        return text

    # Step 2: Construct a TF-IDF matrix for the sentences
    vectorizer = TfidfVectorizer()
    tfidf_matrix = vectorizer.fit_transform(sentences)
```



```
    # Step 3: Apply Singular Value Decomposition (SVD)
    svd = TruncatedSVD(n_components=num_sentences)
    svd_matrix = svd.fit_transform(tfidf_matrix.T)

    # Step 4: Rank the sentences based on their importance
    sentence_scores = svd_matrix.sum(axis=0)
    ranked_sentence_indices = np.argsort(sentence_scores)[::-1][:num_sentences]

    # Step 5: Select the top-ranked sentences to form the summary
    summary = [sentences[i] for i in sorted(ranked_sentence_indices)]
    return ' '.join(summary)
text = """
Nearly two decades ago, Facebook exploded on college campuses as a site for students to stay in touch. Then came Twitter, where people posted about what they had for breakfast, and Instagram, where friends shared photos to keep up with one another.

Today, Instagram and Facebook feeds are full of ads and sponsored posts. TikTok and Snapchat are stuffed with videos from influencers promoting dish soaps and dating apps. And soon, Twitter posts that gain the most visibility will come mostly from subscribers who pay for the exposure and other perks.

Social media is, in many ways, becoming less social. The kinds of posts where people update friends and family about their lives have become harder to see over the years as the biggest sites have become increasingly "corporatized." Instead of seeing messages and photos from friends and relatives about their holidays or fancy dinners, users of Instagram, Facebook, TikTok, Twitter and Snapchat now often view professionalized content from brands, influencers and others that pay for placement.
"""
summary = summarize_text_lsa(text)
print(summary)
```

```
Nearly two decades ago, Facebook exploded on college campuses as a site for students to stay in touch. Then came Twitter,
where people posted about what they had for breakfast, and Instagram, where friends shared photos to keep up with one anot
her. Today, Instagram and Facebook feeds are full of ads and sponsored posts.
```

By taking 20 paragraphs.



```
        Model   Metric  Precision    Recall  F1 Score
0          T5  ROUGE-1   0.117647  0.416667  0.183486
1          T5  ROUGE-2   0.035714  0.130435  0.056075
2          T5  ROUGE-L   0.094118  0.333333  0.146789
3     Pegasus  ROUGE-1   0.054054  0.083333  0.065574
4     Pegasus  ROUGE-2   0.000000  0.000000  0.000000
5     Pegasus  ROUGE-L   0.054054  0.083333  0.065574
6        BRET  ROUGE-1   0.113990  0.916667  0.202765
7        BRET  ROUGE-2   0.062500  0.521739  0.111628
8        BRET  ROUGE-L   0.098446  0.791667  0.175115
9    TextRank  ROUGE-1   0.113990  0.916667  0.202765
10   TextRank  ROUGE-2   0.062500  0.521739  0.111628
11   TextRank  ROUGE-L   0.098446  0.791667  0.175115
12     TF-IDF  ROUGE-1   0.113990  0.916667  0.202765
13     TF-IDF  ROUGE-2   0.062500  0.521739  0.111628
14     TF-IDF  ROUGE-L   0.098446  0.791667  0.175115
15        LSA  ROUGE-1   0.113990  0.916667  0.202765
16        LSA  ROUGE-2   0.062500  0.521739  0.111628
17        LSA  ROUGE-L   0.098446  0.791667  0.175115
```

9.Conclusion

Evaluation of Techniques: You compared traditional extractive methods (TF-IDF, TextRank, LSA) with advanced abstractive models (T5, Pegasus, BERT).
Dataset and Metrics: You used a recognized dataset and standard metrics (ROUGE-1, ROUGE-2, ROUGE-L) to assess performance.
Findings: Transformer-based models (T5, Pegasus, BERT) significantly outperform extractive methods in generating summaries with coherence and contextual relevance. This is due to their ability to capture complex language patterns through deep learning and pre-trained models.
Pegasus model performed better comparatively and the issue of coherence has also been solved by parameter tuning method.
Trade-off: While superior, transformer models require more computational power and training time.

10.References

AAKASH SRIVASTAVA1 , KAMAL CHAUHAN2 , HIMANSHU DAHARWAL3 , NIKHIL MUKATI4 , PRANOTI SHRIKANT KAVIMANDAN5 Department of Computer Science and

Business System Bharati Vidyapeeth Deemed University College of Engineering, Pune, **Text Summarizer Using NLP (Natural Language Processing)** (JUL 2022)

Dr Sumathi Pawara* , Dr Manjula Gururaj Hb , Dr Niranajan N Chiplunarc bAssociate Professor,Nitte(Deemed to be University),NMAMIT, Karkala, Karnataka, India ,cProfessor, Nitte (Deemed to be University),NMAMIT, Karkala, Karnataka, India, **Text Summarization Using Document and Sentence Clustering Method** (2022)

Narendra Andhale Department of Computer Engineering Vishwakarma Institute of Information Technology Pune, India and L.A. Bewoor Department of Computer Engineering Vishwakarma Institute of Information Technology Pune, India .**An Overview of Text Summarization Techniques**

G. Karuna1* , M. Akshith1 , Parige Sai Dinesh1 , Bodhan Vishnu Vardhan1 , Yashwant Singh Bisht2 , M. N. Narsaiah,1 Department of CSE (AIML), GRIET, Hyderabad, Telangana State, India 2Uttaranchal Institute of Technology, Uttaranchal University, Dehradun, 248007, India 3KG Reddy College of Engineering & Technology, Hyderabad, India,**Automated Abstractive Text Summarization using Deep Learning.**E3S Web of Conferences(2023)