Infosys®
Navigate your next

Infosys
Springboard

TEXT SUMMARIZATION

Mentor : Mr. Narendra kumar

Intern: Patel Hetu

B.Tech 3rd year(ICT)

Group-1

Date : 3-7-2024

INTRODUCTION

•**Definition of Text Summarization:**

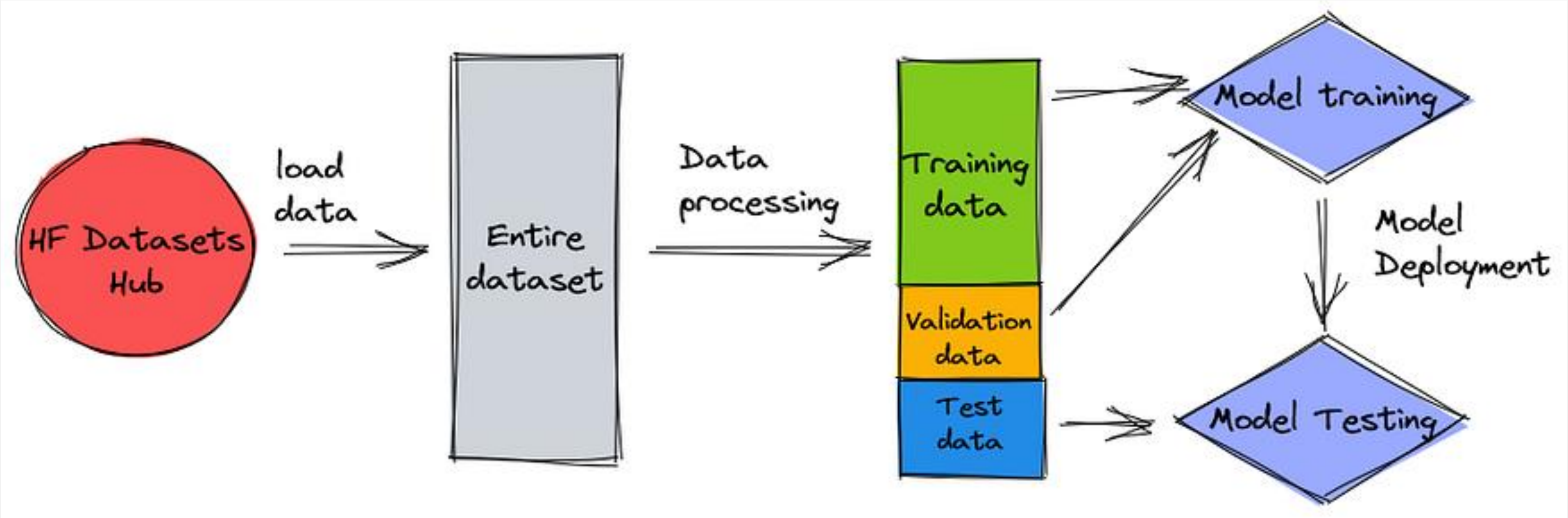
•"Text summarization is the process of creating a concise and coherent version of a longer text document.“

•**Importance:**
•"Summarization helps quickly understand large volumes of information.“

•**Applications:**
•"Used in news articles, research papers, legal documents, customer feedback, and social media."

Used datasets:
CNN/daily mails(extractive part)
Samsum(abstractive part)

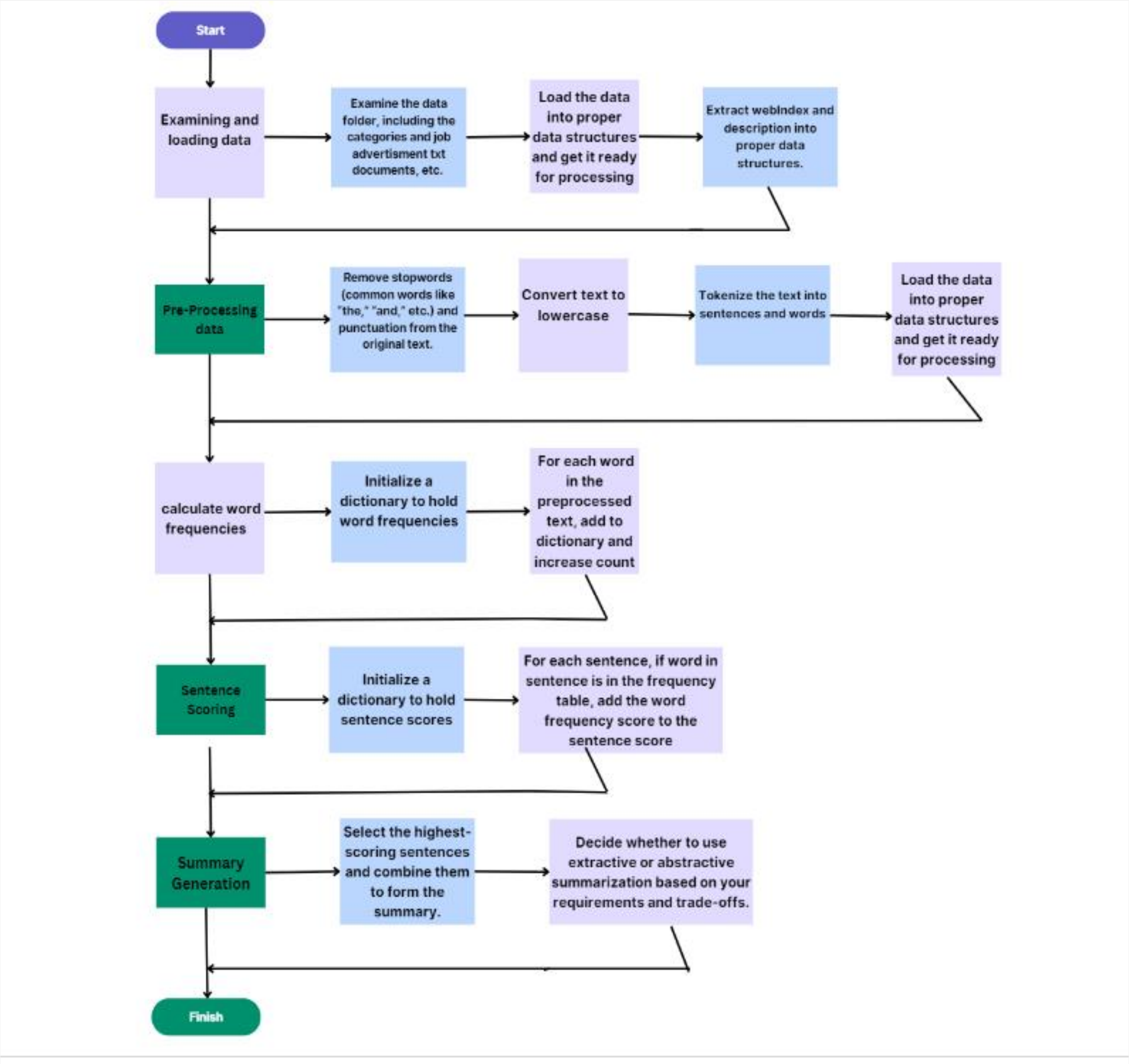
INTENDED PLAN:



TECHNIQUES AND ALGORITHMS

- **Basic Techniques:**
- Frequency-based methods: Select sentences based on word frequency.
- TF-IDF: Measures the importance of words in a document relative to a corpus.
- Sentence scoring and ranking: Ranks sentences based on various metrics.
- **Advanced Algorithms:**
- **LexRank and TextRank:** Graph-based algorithms that rank sentences based on their importance.
- **Latent Semantic Analysis (LSA):** Uses singular value decomposition to identify important concepts.
- **Neural Networks and Deep Learning:**
  - RNNs: Sequence processing neural networks.
  - LSTMs: Advanced RNNs capable of learning long-term dependencies.
  - Attention Mechanisms: Enhance the focus on relevant parts of the input.
- **Transformer Models:**
  - BERT: A model that understands context in both directions.
  - GPT: A generative model for producing coherent text.
  - T5: A model that treats all tasks as text-to-text transformations.

Work flow:



Start

Examining and loading data

Examine the data folder, including the categories and job advertisment txt documents, etc.

Load the data into proper data structures and get it ready for processing

Extract webIndex and description into proper data structures.

Pre-Processing data

Remove stopwords (common words like "the," "and," etc.) and punctuation from the original text.

Convert text to lowercase

Tokenize the text into sentences and words

Load the data into proper data structures and get it ready for processing

calculate word frequencies

Initialize a dictionary to hold word frequencies

For each word in the preprocessed text, add to dictionary and increase count

Sentence Scoring

Initialize a dictionary to hold sentence scores

For each sentence, if word in sentence is in the frequency table, add the word frequency score to the sentence score

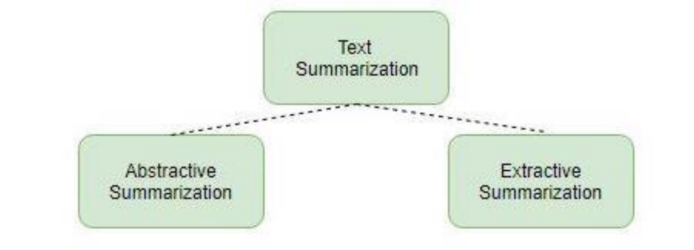
Summary Generation

Select the highest-scoring sentences and combine them to form the summary.

Decide whether to use extractive or abstractive summarization based on your requirements and trade-offs.

Finish

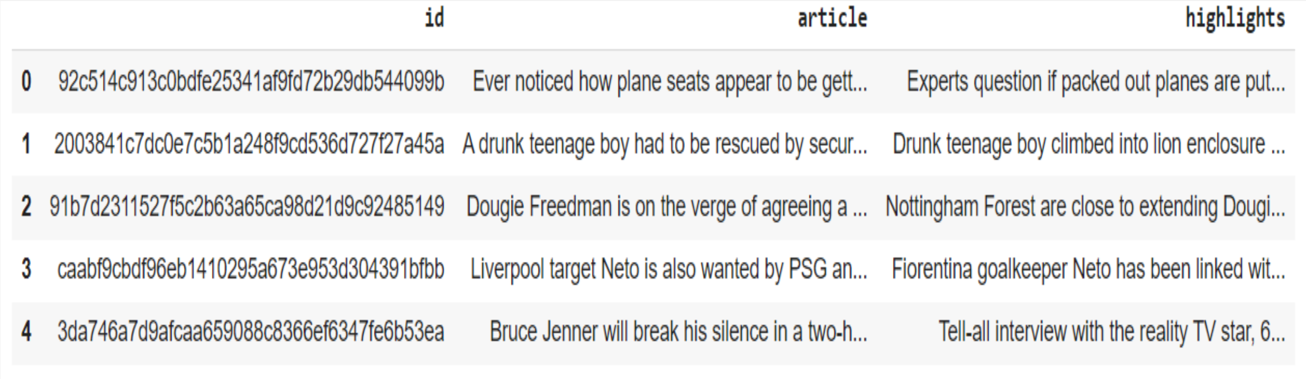
TYPES OF TEXT SUMMARIZATION



- **Extractive Summarization:**
- Extractive summarization selects key sentences, phrases, or sections directly from the source text.
- It often uses methods like frequency-based selection and TF-IDF.
- Pros: Simple and quick.
- Cons: May lack coherence and context.
- **Abstractive Summarization:**
- Abstractive summarization generates new sentences that convey the main ideas of the source text.
- It involves complex techniques like neural networks and transformer models.
- Pros: Can produce more coherent and human-like summaries.
- Cons: More computationally intensive and may introduce inaccuracies.

DATASET PROPOSAL:

- Merged dataset from
- CNN/Daily mails
- Samsum



	id	article	highlights
0	92c514c913c0bdfe25341af9fd72b29db544099b	Ever noticed how plane seats appear to be gett...	Experts question if packed out planes are put...
1	2003841c7dc0e7c5b1a248f9cd536d727f27a45a	A drunk teenage boy had to be rescued by secur...	Drunk teenage boy climbed into lion enclosure ...
2	91b7d2311527f5c2b63a65ca98d21d9c92485149	Dougie Freedman is on the verge of agreeing a ...	Nottingham Forest are close to extending Dougi...
3	caabf9cbdf96eb1410295a673e953d304391bfbb	Liverpool target Neto is also wanted by PSG an...	Fiorentina goalkeeper Neto has been linked wit...
4	3da746a7d9afcaa659088c8366ef6347fe6b53ea	Bruce Jenner will break his silence in a two-h...	Tell-all interview with the reality TV star, 6...

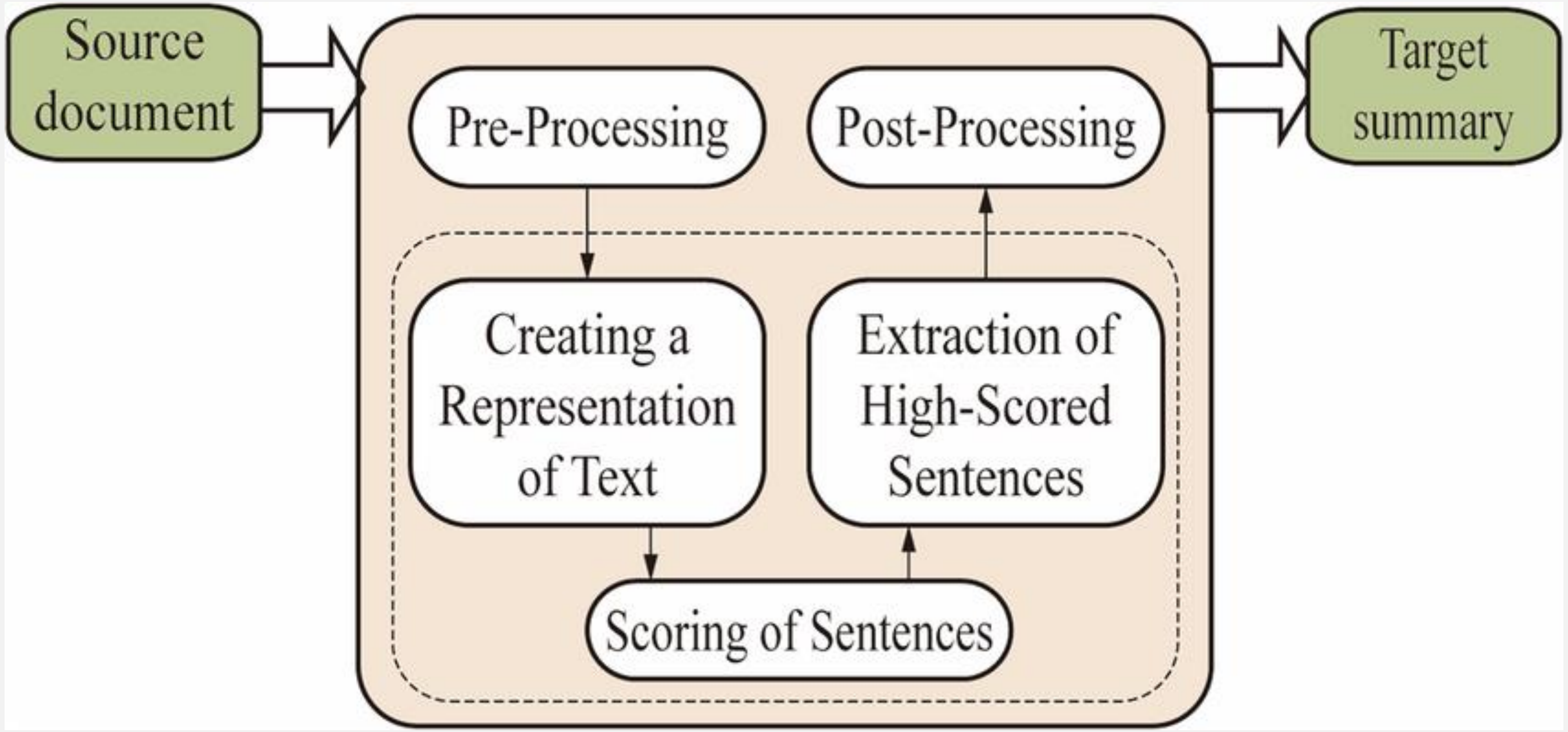
EXTRACTIVE SUMMARIZATION

•**Definition:**
•"Extractive summarization involves selecting and concatenating the most important sentences from the source text.“

•**Methods:**
•"Frequency-based selection, TF-IDF.“

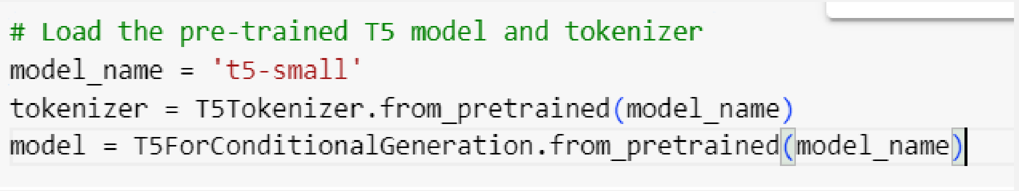
•**Pros and Cons:**
•"Pros: Simple and quick."
•"Cons: May lack coherence and context."

ARCHITECTURE OF THE EXTRACTIVE TEXT SUMMARIZATION SYSTEM:



ROUGE SCORE

-
  ```
  'rouge1': AggregateScore(low=Score(precision=0.75, recall=0.6,
  fmeasure=0.6666666666666665), mid=Score(precision=0.75, recall=0.6,
  fmeasure=0.6666666666666665), high=Score(precision=0.75, recall=0.6,
  fmeasure=0.6666666666666665)), 'rouge2':
  AggregateScore(low=Score(precision=0.3333333333333333, recall=0.25,
  fmeasure=0.28571428571428575), mid=Score(precision=0.3333333333333333,
  recall=0.25, fmeasure=0.28571428571428575),
  high=Score(precision=0.3333333333333333, recall=0.25,
  fmeasure=0.28571428571428575)), 'rougeL':
  AggregateScore(low=Score(precision=0.75, recall=0.6,
  fmeasure=0.6666666666666665), mid=Score(precision=0.75, recall=0.6,
  fmeasure=0.6666666666666665), high=Score(precision=0.75, recall=0.6,
  fmeasure=0.6666666666666665)), 'rougeLsum':
  AggregateScore(low=Score(precision=0.75, recall=0.6,
  fmeasure=0.6666666666666665), mid=Score(precision=0.75, recall=0.6,
  fmeasure=0.6666666666666665), high=Score(precision=0.75, recall=0.6,
  fmeasure=0.6666666666666665))
  ```

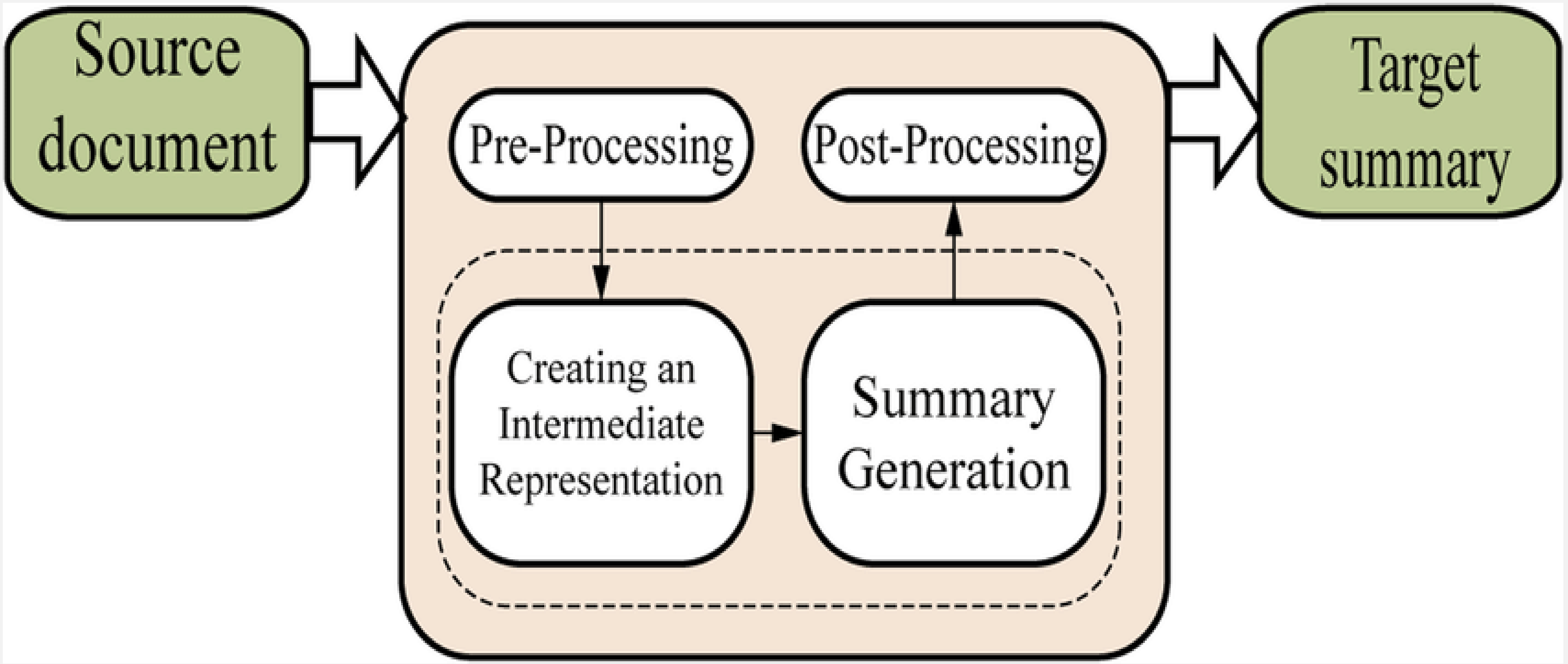


```
# Load the pre-trained T5 model and tokenizer
model_name = 't5-small'
tokenizer = T5Tokenizer.from_pretrained(model_name)
model = T5ForConditionalGeneration.from_pretrained(model_name)
```

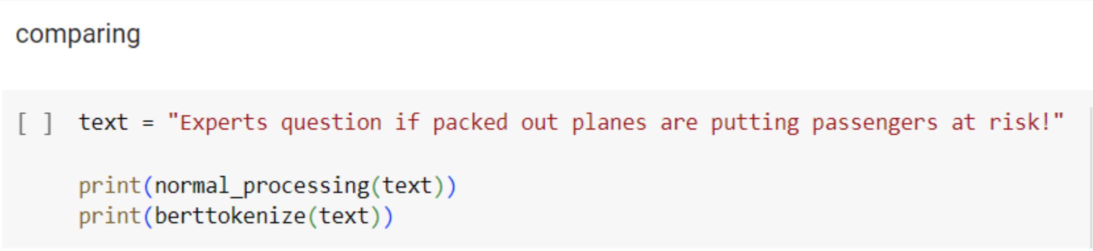
ABSTRACTIVE SUMMARIZATION

- **Definition:**
  - "Abstractive summarization generates new sentences to represent the core ideas of the text."
- **Techniques:**
  - "Uses neural networks and transformer models."
- **Pros and Cons:**
  - "Pros: More coherent and human-like summaries."
  - "Cons: Computationally intensive and may introduce inaccuracies."

ARCHITECTURE OF THE ABSTRACTIVE TEXT SUMMARIZATION SYSTEM:



COMPARING MODELS

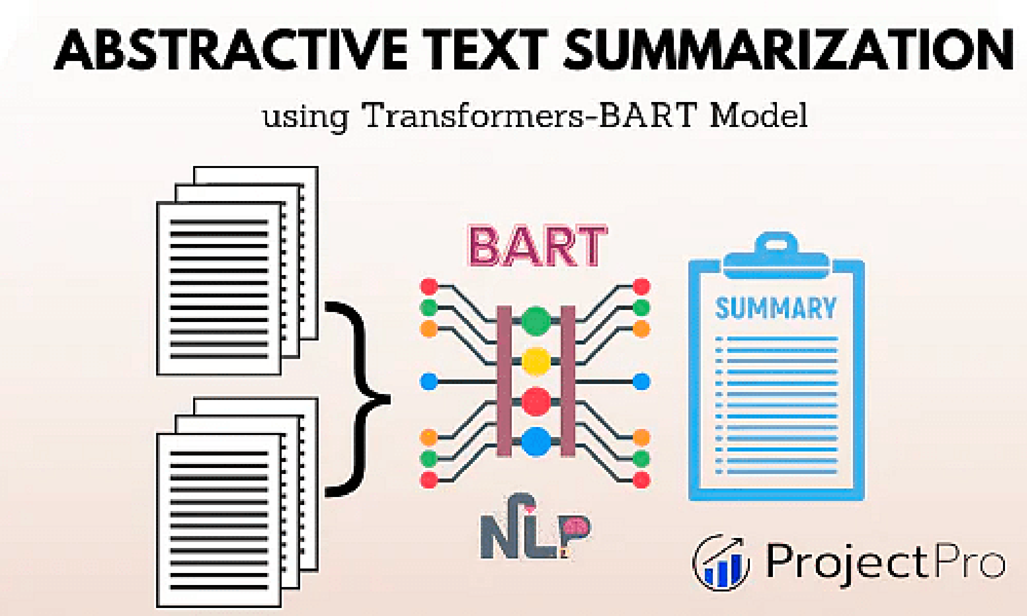


comparing

```
[ ]  text = "Experts question if packed out planes are putting passengers at risk!"

     print(normal_processing(text))
     print(berttokenize(text))
```

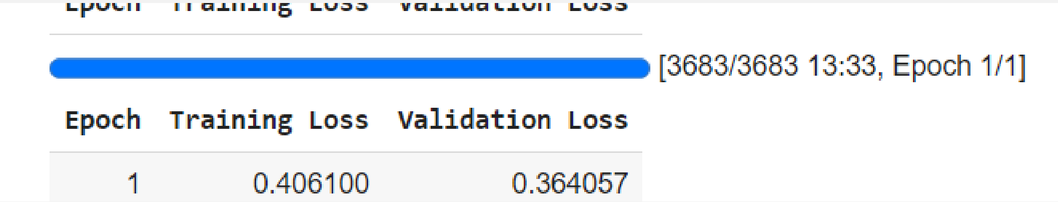
OUTPUT:

```
Normal processing Experts question if packed out planes are
putting passengers at risk! ['expert', 'question', 'pack',
'plane', 'put', 'passeng', 'risk'] {'input_ids': [101,
8519, 3160, 2065, 8966, 2041, 9738, 2024, 5128, 5467, 2012,
3891, 999, 102], 'token_type_ids': [0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0], 'attention_mask': [1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1]}
```



MODEL TRAINING :

```
TrainOutput(global_step=3683,
training_loss=0.5255669773256134,
metrics={'train_runtime': 814.185,
'train_samples_per_second': 18.094,
'train_steps_per_second': 4.524, 'total_flos':
1993855419285504.0, 'train_loss': 0.5255669773256134,
'epoch': 1.0})
```



Epoch Training Loss Validation Loss

[3683/3683 13:33, Epoch 1/1]

Epoch	Training Loss	Validation Loss
1	0.406100	0.364057

EVALUATE THE MODEL(MODEL VALIDATION)

- Evaluate the model using the ROUGE metric.
- ROUGE-1 (R1): ROUGE-1 measures the overlap of unigram (single word) tokens between the generated summary and the reference (gold-standard) summary. It calculates the precision, recall, and F1 score of unigrams.
- ROUGE-2 (R2): ROUGE-2 measures the overlap of bigram (two-word sequences) tokens between the generated summary and the reference summary. Similar to ROUGE-1, it calculates precision, recall, and F1 score of bigrams.
- ROUGE-L (RL): ROUGE-L measures the longest common subsequence (LCS) between the generated summary and the reference summary. It calculates precision, recall, and F1 score based on the length of the LCS.
- ROUGE-W (RW): ROUGE-W (sometimes referred to as ROUGE-Lsum) measures the weighted LCS between the generated summary and the reference summary. It assigns more weight to longer matches in the LCS.

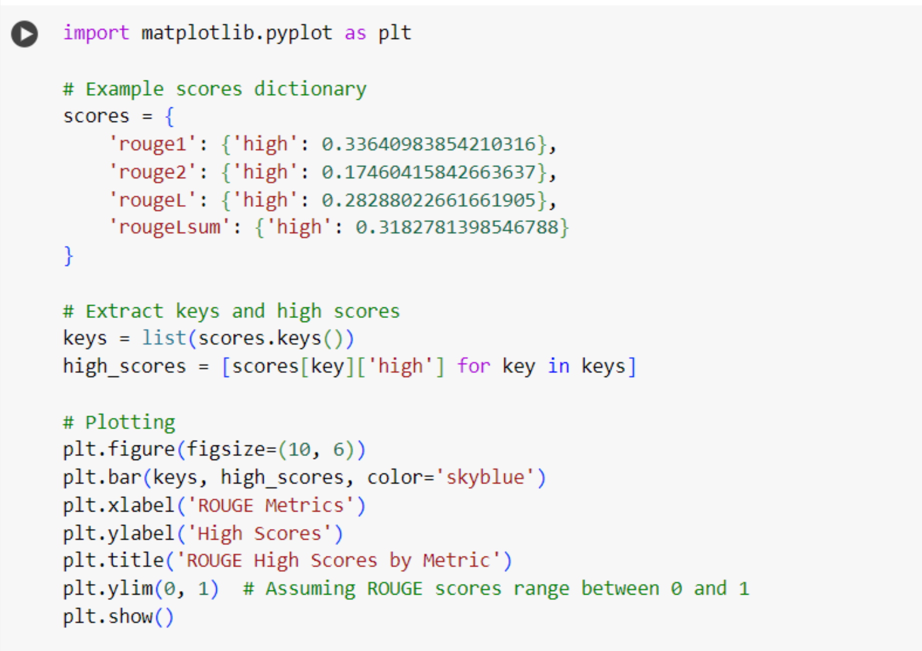
ROUGE SCORE:

```
rouge1: Score(precision=0.767627392778128, recall=0.22941014983341268, fmeasure=0.33640983854210316)
 rouge2: Score(precision=0.41012512677602814, recall=0.11619754543090752, fmeasure=0.17460415842663637)
rougeL: Score(precision=0.5740780603041631, recall=0.19681507337121906, fmeasure=0.2828802261661905)
rougeLsum: Score(precision=0.7189119882750398, recall=0.21573804703761962, fmeasure=0.3182781398546788)
```

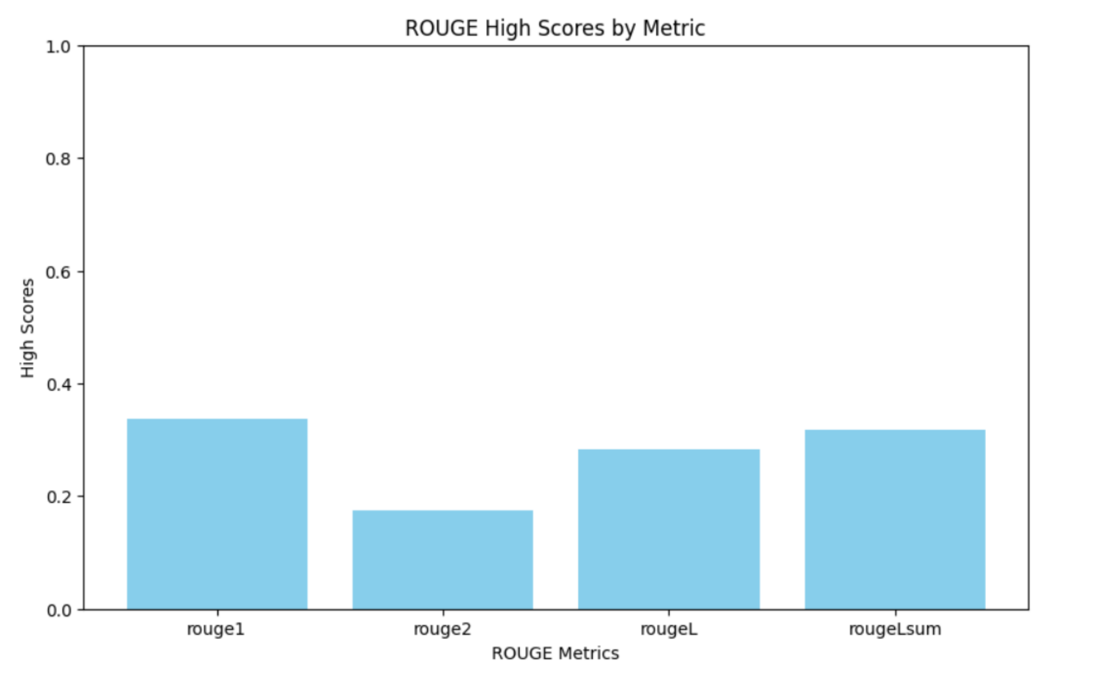
https://github.com/Patelhlt/text-summarizer/blob/main/Interface.ipynb

Fine-tuned-Abstractive:
https://drive.google.com/drive/folders/1w2cE6bqU-YomgUloOUafl7zOatt287OY?usp=sharing

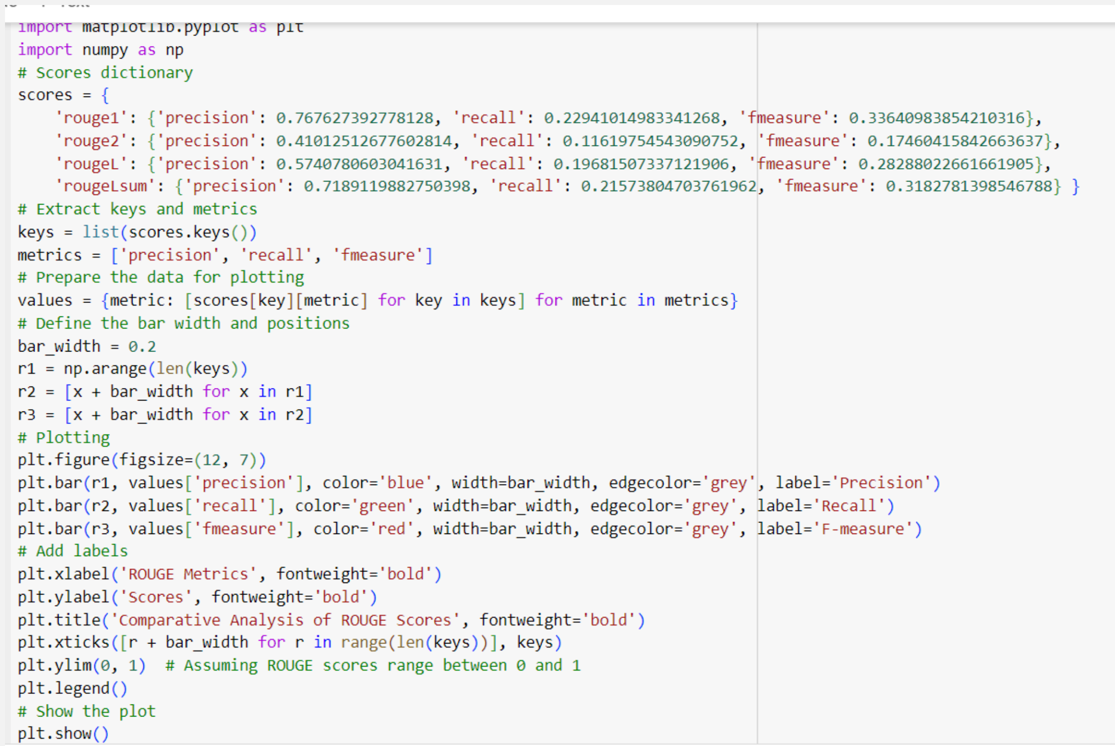
PLOTTING :



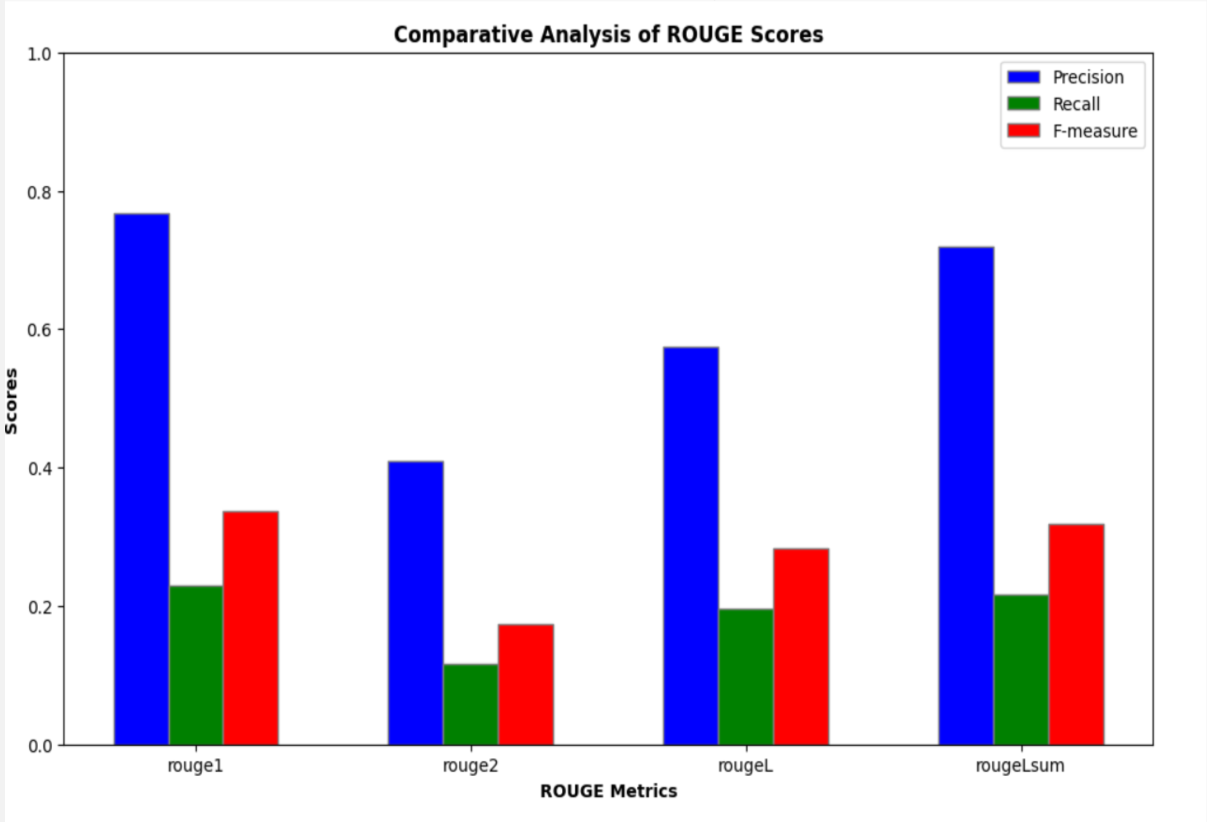
```
import matplotlib.pyplot as plt

# Example scores dictionary
scores = {
    'rouge1': {'high': 0.33640983854210316},
    'rouge2': {'high': 0.17460415842663637},
    'rougeL': {'high': 0.28288022661661905},
    'rougeLsum': {'high': 0.3182781398546788}
}

# Extract keys and high scores
keys = list(scores.keys())
high_scores = [scores[key]['high'] for key in keys]

# Plotting
plt.figure(figsize=(10, 6))
plt.bar(keys, high_scores, color='skyblue')
plt.xlabel('ROUGE Metrics')
plt.ylabel('High Scores')
plt.title('ROUGE High Scores by Metric')
plt.ylim(0, 1)  # Assuming ROUGE scores range between 0 and 1
plt.show()
```



COMPARATIVE ANALYSIS:



```
import matplotlib.pyplot as plt
import numpy as np
# Scores dictionary
scores = {
    'rouge1': {'precision': 0.767627392778128, 'recall': 0.22941014983341268, 'fmeasure': 0.33640983854210316},
    'rouge2': {'precision': 0.41012512677602814, 'recall': 0.11619754543090752, 'fmeasure': 0.17460415842663637},
    'rougeL': {'precision': 0.5740780603041631, 'recall': 0.19681507337121906, 'fmeasure': 0.28288022661661905},
    'rougeLsum': {'precision': 0.7189119882750398, 'recall': 0.21573804703761962, 'fmeasure': 0.3182781398546788} }
# Extract keys and metrics
keys = list(scores.keys())
metrics = ['precision', 'recall', 'fmeasure']
# Prepare the data for plotting
values = {metric: [scores[key][metric] for key in keys] for metric in metrics}
# Define the bar width and positions
bar_width = 0.2
r1 = np.arange(len(keys))
r2 = [x + bar_width for x in r1]
r3 = [x + bar_width for x in r2]
# Plotting
plt.figure(figsize=(12, 7))
plt.bar(r1, values['precision'], color='blue', width=bar_width, edgecolor='grey', label='Precision')
plt.bar(r2, values['recall'], color='green', width=bar_width, edgecolor='grey', label='Recall')
plt.bar(r3, values['fmeasure'], color='red', width=bar_width, edgecolor='grey', label='F-measure')
# Add labels
plt.xlabel('ROUGE Metrics', fontweight='bold')
plt.ylabel('Scores', fontweight='bold')
plt.title('Comparative Analysis of ROUGE Scores', fontweight='bold')
plt.xticks([r + bar_width for r in range(len(keys))], keys)
plt.ylim(0, 1)  # Assuming ROUGE scores range between 0 and 1
plt.legend()
# Show the plot
plt.show()
```



INTERFACE :

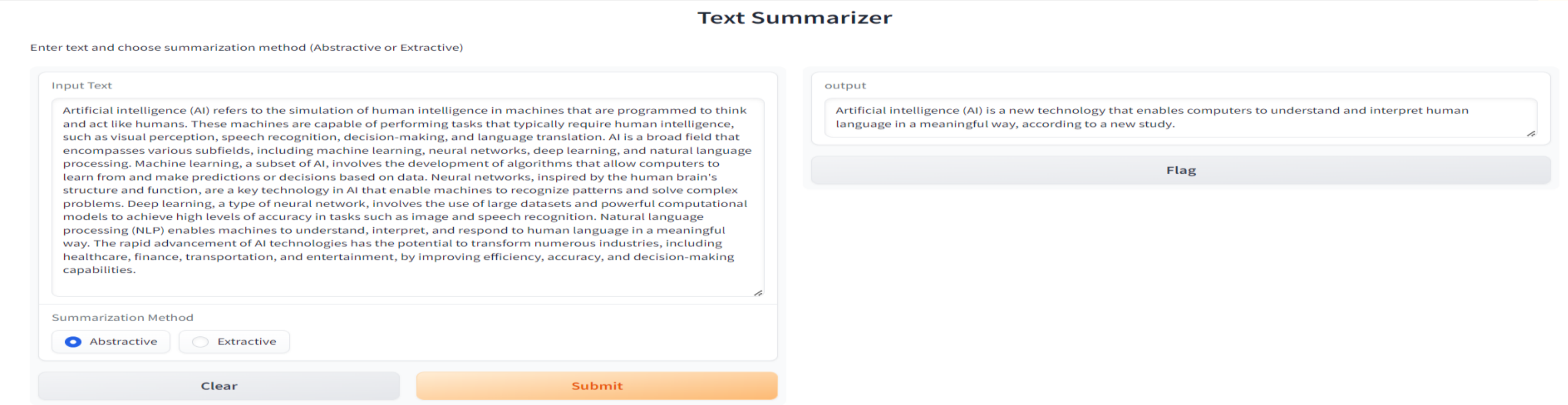
Components of a Text Summarization Interface

1. **Input Field**:
    1. **Text Box**: A large text box where users can paste or type the text they want to summarize.
    2. **File Upload**: Option to upload text files (e.g., .txt, .docx, .pdf) for summarization.
2. **Summarization Options**:
    1. **Type of Summarization**: Choose between extractive (selects important sentences from the original text) and abstractive (generates new sentences that convey the main ideas) summarization.
    2. **Summary Length**: Slider or input box to specify the desired length or percentage of the summary.
3. **Output Display**:
    1. **Summary Box**: A text box or display area where the summarized text is shown.
    2. **Download Option**: Button to download the summary as a text file.
4. **Additional Features**:
    1. **Language Selection**: Option to select the language for summarization.
    2. **Adjustable Parameters**: Advanced settings for adjusting parameters like temperature, beam size, or max tokens for abstractive models.
5. **User Feedback**:
    1. **Edit and Improve**: Option for users to manually edit the generated summary.
    2. **Feedback Form**: Collect user feedback to improve the summarization model.

Design Considerations

1. **User Experience (UX)**:
    1. **Simplicity**: Keep the interface clean and intuitive.
    2. **Responsiveness**: Ensure the interface works well on various devices, including desktops, tablets, and smartphones.
2. **Performance**:
    1. **Speed**: The summarization process should be fast to enhance user satisfaction.
    2. **Scalability**: The system should handle multiple requests efficiently.
3. **Accuracy and Quality**:
    1. **Model Selection**: Use state-of-the-art models for better summarization quality.
    2. **Continuous Improvement**: Regularly update the models based on user feedback and advancements in NLP.
4. **Security and Privacy**:
    1. **Data Privacy**: Ensure that user data is not stored or misused.
    2. **Secure Uploads**: Implement secure file handling practices.

Testing interface using gradio:



Text Summarizer

Enter text and choose summarization method (Abstractive or Extractive)

Input Text

Artificial intelligence (AI) refers to the simulation of human intelligence in machines that are programmed to think and act like humans. These machines are capable of performing tasks that typically require human intelligence, such as visual perception, speech recognition, decision-making, and language translation. AI is a broad field that encompasses various subfields, including machine learning, neural networks, deep learning, and natural language processing. Machine learning, a subset of AI, involves the development of algorithms that allow computers to learn from and make predictions or decisions based on data. Neural networks, inspired by the human brain's structure and function, are a key technology in AI that enable machines to recognize patterns and solve complex problems. Deep learning, a type of neural network, involves the use of large datasets and powerful computational models to achieve high levels of accuracy in tasks such as image and speech recognition. Natural language processing (NLP) enables machines to understand, interpret, and respond to human language in a meaningful way. The rapid advancement of AI technologies has the potential to transform numerous industries, including healthcare, finance, transportation, and entertainment, by improving efficiency, accuracy, and decision-making capabilities.

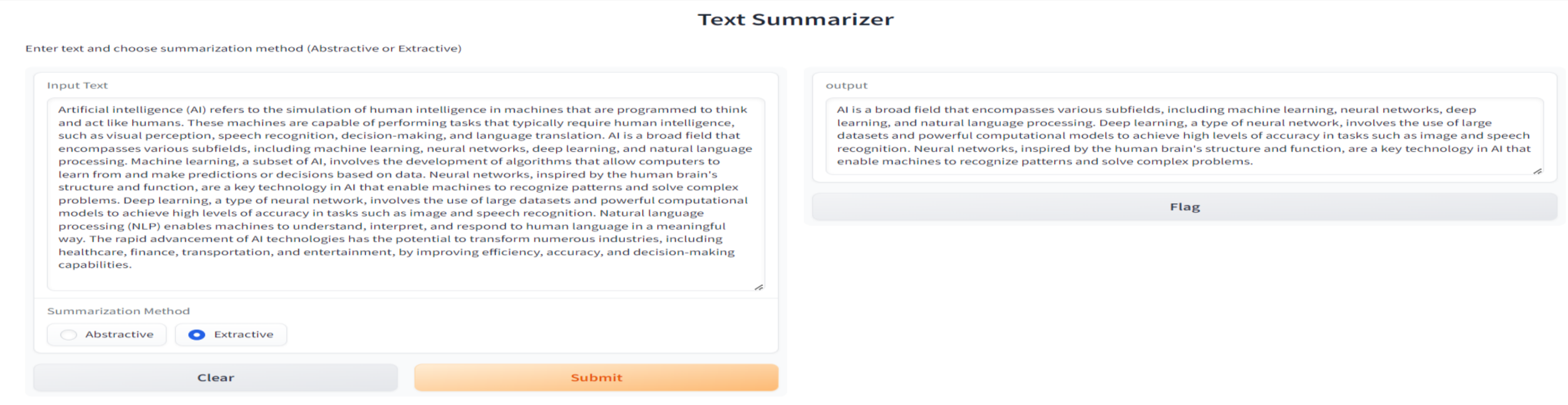
Summarization Method

● Abstractive ○ Extractive

output

Artificial intelligence (AI) is a new technology that enables computers to understand and interpret human language in a meaningful way, according to a new study.

fig: abstractive model



Text Summarizer

Enter text and choose summarization method (Abstractive or Extractive)

Input Text

Artificial intelligence (AI) refers to the simulation of human intelligence in machines that are programmed to think and act like humans. These machines are capable of performing tasks that typically require human intelligence, such as visual perception, speech recognition, decision-making, and language translation. AI is a broad field that encompasses various subfields, including machine learning, neural networks, deep learning, and natural language processing. Machine learning, a subset of AI, involves the development of algorithms that allow computers to learn from and make predictions or decisions based on data. Neural networks, inspired by the human brain's structure and function, are a key technology in AI that enable machines to recognize patterns and solve complex problems. Deep learning, a type of neural network, involves the use of large datasets and powerful computational models to achieve high levels of accuracy in tasks such as image and speech recognition. Natural language processing (NLP) enables machines to understand, interpret, and respond to human language in a meaningful way. The rapid advancement of AI technologies has the potential to transform numerous industries, including healthcare, finance, transportation, and entertainment, by improving efficiency, accuracy, and decision-making capabilities.

Summarization Method

○ Abstractive ● Extractive

output

AI is a broad field that encompasses various subfields, including machine learning, neural networks, deep learning, and natural language processing. Deep learning, a type of neural network, involves the use of large datasets and powerful computational models to achieve high levels of accuracy in tasks such as image and speech recognition. Neural networks, inspired by the human brain's structure and function, are a key technology in AI that enable machines to recognize patterns and solve complex problems.

fig: extractive model

https://drive.google.com/drive/folders/1w2cE6bqUYomgUloOUafl7zOatt287OY?usp=sharing

DEPLOYMENT PART:

POPULAR PLATFORMS

1. Hugging Face Spaces

About: Hugging Face Spaces is a hosted platform that allows you to deploy machine learning models and applications. It's especially well-suited for models developed with popular frameworks like Gradio, Streamlit, and others. Spaces provide easy integration, seamless deployment, and a community-driven ecosystem.

Pros:

- Easy to use and integrate with Hugging Face models.
- Free tier available.
- Supports Gradio and other web app frameworks.
- No need for server management.

Cons:

- Limited compute resources in the free tier.
- May require knowledge of Hugging Face ecosystem.

Website: Hugging Face Spaces

2. Heroku

About: Heroku is a cloud platform as a service (PaaS) that enables developers to build, run, and oper cloud. It supports multiple programming languages and frameworks, including Python and Gradio.

Pros:

- Easy deployment with Git.
- Free tier available with limitations.
- Supports a wide range of programming languages and frameworks.

Cons:

- Limited free tier with dyno sleeping and restricted resources.
- May require additional setup for machine learning models.

Website: Heroku

POPULAR PLATFORMS

3. Google Cloud Platform (GCP)

About: Google Cloud Platform (GCP) provides a suite of cloud computing services that run on the same infrastructure that Google uses internally. It offers various services for deploying machine learning models, including AI Platform, App Engine, and Cloud Run.

Pros:

- Robust and scalable infrastructure.
- Wide range of services tailored for machine learning and AI.
- Integration with other Google services.

Cons:

- Steeper learning curve compared to other options.
- Costs can accumulate quickly without proper management.

Website: [Google Cloud Platform]

4. Amazon Web Services (AWS)

About: AWS is a comprehensive and widely adopted cloud platform offering over 200 fully featured services from data centers globally. For machine learning deployments, AWS provides services like SageMaker, Lambda, and EC2.

Pros:

- Highly scalable and reliable.
- Comprehensive suite of tools and services.
- Strong community and support.

Cons:

- Can be complex to navigate and set up.
- Costs can be high, especially for large-scale deployments.

Website: [Amazon Web Services]

TOOLS AND LIBRARIES

- **Spacy:**
  - "A fast and efficient NLP library."
  - "Useful for preprocessing and entity recognition, aiding summarization tasks."
- **Gensim:(optional)**
  - "A robust library for topic modeling and document similarity analysis."
  - "Implements algorithms like TextRank and LSA for summarization."
- **Hugging Face Transformers:**
  - "A library offering pre-trained transformer models.“
  - "Enables easy implementation of state-of-the-art summarization models.

CASE STUDIES AND EXAMPLES

- **Real-world Applications:**
  - "Summarizing news articles for concise updates."
  - "Condensing research papers for quick review."
  - "Creating summaries of legal documents."
- **Detailed Examples:**
  - "Input texts and their summaries."
  - "Comparison between extractive and abstractive methods."
  - "Code snippets demonstrating the use of various tools and libraries."

CHALLENGES AND FUTURE DIRECTIONS

- **Current Limitations:**
  - "Difficulty in understanding context and maintaining coherence."
  - "Handling diverse and large datasets."
  - "Issues with factual accuracy and grammar in abstractive summarization."
- **Future Trends:**
  - "Improvements in deep learning and transformer models."
  - "Integration with other AI technologies like chatbots."
  - "Development of better evaluation metrics for summarization quality."

CONCLUSION AND REFERENCES

- **Summary:**
- "Text summarization is a crucial tool for managing information overload."
- "Both extractive and abstractive methods have their uses and challenges."
- "Ongoing research is vital for improving summarization techniques.“

- **References:**
- "List of academic papers, books, articles, and online resources cited.“
- “Kaggle”
- ChatGPT, perplexity.ai etc…