A STUDY ON ABSTRACTIVE TEXT SUMMARIZATION

Internship Project Report

Submitted by

U. SURIYA

(Register No: 820421205073)

Under the guidance of

Dr. K. NANDHINI, M.Tech., Ph.D.,

Assistant Professor

Department of Computer Science

Central University of Tamil Nadu, Thiruvarur

AUGUST-2024

DECLARATION

I am **U. SURIYA** (Register No: 820421205073), hereby declare that this internship project report entitled "**A STUDY ON ABSTRACTIVE TEXT SUMMARIZATION**" submitted to the Department of Computer Science, Central University of Tamil Nadu, Thiruvarur, a record of summer INTERNSHIP project work done by me from **(31-07-2024 to 30-08-2024)**., under the guidance of **Dr. K. NANDHINI, M.Tech., Ph.D.**, Assistant Professor, Department of Computer Science, Central University of Tamil Nadu, Thiruvarur.

Place: Thiruvarur (U. SURIYA)

Date:

ACKNOWLEDGEMENT

I would like to sincerely thank Vice-Chancellor **Prof. M. Krishnan** and **Prof. R.Thirumurugan**, Registrar(I/c) of Central University of Tamil Nadu, Thiruvarur extending administrative permission for this internship.

I would like to express my sincere gratitude to **Dr. K. Nandhini,** my internship supervisor, for her invaluable guidance and support throughout my internship. From the moment I started, she took the time to get to know me and understand my goals for the internship. Her clear direction, expectations, and availability to answer my questions were instrumental in my professional growth. Her constructive feedback helped me improve my skills, and her encouragement kept me motivated. I am deeply thankful for her commitment to my success.

I would like to thank **Dr. Chandramouli P.V.S.S.R**, Head, Department of Computer Science, Central University of Tamil Nadu, Thiruvarur, for providing permission to do this summer internship project.

I would also like to thank my colleagues and the staff at the Central University of Tamil Nadu for their warm welcome and assistance during my internship. Working with such a talented and supportive team was a pleasure. My colleagues willingly shared their knowledge, and the staff efficiently assisted me with any needs or questions I had.

I also highly indebted to extend my sincere thanks to my department **Dr. K. Mohan, Headof the Department**, Faculties and Principal of Anjalai Ammal Mahalingam Engineering College, Kovilvenni, Thiruvarur, who permitted me to do this internship project at CUTN

Once again, thank you to everyone who contributed to making my internship a valuable experience.

U. Suriya

Final year (820421205073)

B. Tech. IT

Anjalai Ammal Mahalingam Engineering College

Kovilvenni, Thiruvarur

INTERNSHIP CERTIFICATE

ABSTRACT

In this project, Automatic text summarization is basically summarizing of the given paragraph using natural language processing and machine learning. There has been an explosion in the amount of text data from a variety of sources. This volume of text is an invaluable source of information and knowledge which needs to be effectively summarized to be useful. In this review, the main approaches to automatic text summarization are described. We review the different processes for summarization and describe the effectiveness and shortcomings of the different methods. The system works by assigning scores to sentences in the document to be summarized and using the highest scoring sentences in the summary.

TABLE OF CONTENTS

S.NO	TITLE	PAGE.NO
1	LIST OF FIGURES	**7**
2	INTRODUCTION ABOUT TEXT SUMMARIZATION	**8**
3	LITERATURE REVIEW	**10**
4	PROBLEM STATEMENT AND FEASIBILITY STUDY	**11**
5	PEGASUS MODEL	**13**
6	IMPLEMENTATION FOR TEXT SUMMARIZATION IN PEGASUS	**14**
7	FUTURE SCOPE	**18**
8	CONCLUSION	**19**
9	REFERENCES	**20**
10	BIO DATA	**21**

1. LIST OF FIGURES

S. NO	FIG. NO	FIGURE TITLE	PAGE NO
1	1	Illustration of NLP-Driven Text Summarization Process	8
2	2	Types of Text Summarization Approaches	9
3	3	BASE ARCHITECTURE OF PEGASUS MODEL	13
4	4	DEFINING THE CNN/DAILYMAIL DATA ON PEGASUS MODEL	15
5	5	ROUGE SCORE CALCULATION	16
6	6	LOADING SAMSUM DATASET IN PEGASUS MODEL	17
7	7	TEXT SUMMARIZATION RESULT	17

2. INTRODUCTION

TEXT SUMMARIZATION USING NLP

In the modern Internet age, textual data is ever increasing. Need some way to condense this data while preserving the information and meaning. We need to summarize textual data for that.

Text summarization is the process of automatically generating natural language summaries from an input document while retaining the important points. It would help with easy and fast retrieval of information.

As online textual data grows, automatic text summarization methods have the potential to be very helpful because more useful information can be read in a short time.

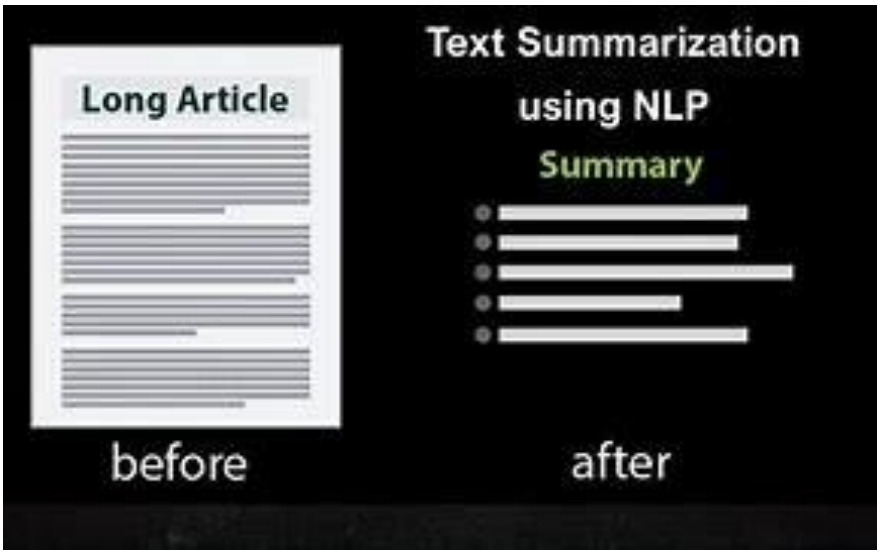

Fig:1-***Illustration of NLP-Driven Text Summarization Process***

Type of summarization:

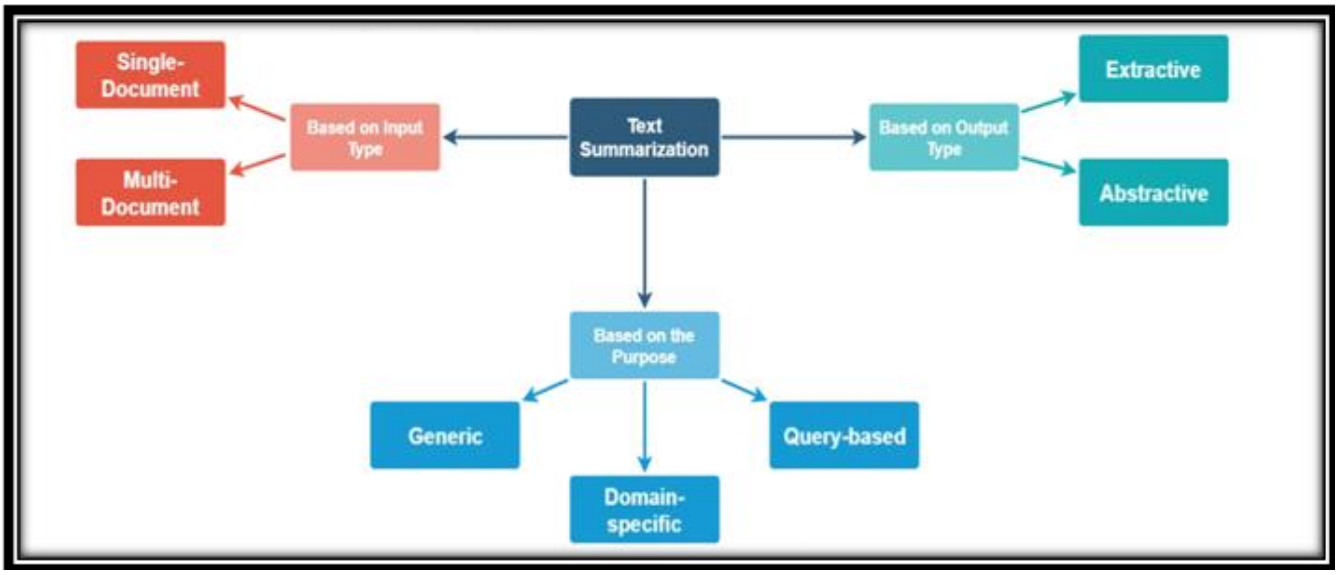

Fig:2-*Types of Text Summarization Approaches*

Based on input type:

1. Single Document, where the input length is short.
2. Multi-Document, where the input can be arbitrarily long.

Based on the purpose:

1. Generic, where the model makes no assumptions about the domain or content of the text to be summarized and treats all inputs as homogeneous.
2. Domain-specific, where the model uses domain-specific knowledge to form a more accurate summary. For example, summarizing research papers of a specific domain, biomedical documents, etc.
3. Query-based, where the summary only contains information that answers natural language questions about the input text.

Based on output type:

1. **Extractive**, where important sentences are selected from the input text to form a summary. Most summarization approaches today are extractive in nature.
2. **Abstractive**, where the model forms its own phrases and sentences to offer a more coherent summary, like what a human would generate.

3. LITERATURE REVIEW

Machine Learning and Statistical Approaches:

Kupiec et al. (1995) introduced machine learning techniques for extractive summarization, leveraging Bayesian classifiers to identify salient sentences, a pivotal step towards statistical summarization models. Nenkova and McKeown (2011) reviewed statistical and machine learning techniques, including supervised and unsupervised methods, providing a comprehensive understanding of how feature selection and model training contribute to summarization performance.

Neural Network Models and Deep Learning:

Rush et al. (2015) marked a significant shift towards deep learning by introducing neural attention mechanisms for abstractive summarization, dramatically improving summary quality and coherence. Chopra et al. (2016) extended this approach with sequence-to-sequence models, demonstrating the effectiveness of encoder-decoder architectures in capturing the essence of lengthy texts. See et al. (2017) introduced pointer-generator networks, which combined the strengths of extractive and abstractive methods, further enhancing summary accuracy and reducing factual errors.

Transformer-Based Models:

Vaswani et al. (2017) revolutionized text summarization with the introduction of the Transformer architecture, enabling more efficient and accurate handling of long-range dependencies in text. Lewis et al. (2020) and Raffel et al. (2020) adapted Transformers for summarization tasks with models like BART and T5, which set new benchmarks in summarization by leveraging bidirectional and autoregressive properties.

Recent Advances and Emerging Trends*:*

Zhang et al. (2020) introduced PEGASUS, a pre-trained model specifically designed for summarization, which employed gap sentences to better understand summarization context and content selection. Xu et al. (2022) explored the integration of reinforcement learning with summarization models, aiming to optimize summary coherence and informativeness beyond traditional training objectives.

4. PROBLEM DEFINITION AND FEASIBILITY ANALYSIS

Problem Definition:

Text Summarization is a crucial task in natural language processing (NLP) that involves condensing long documents into shorter versions while preserving the essential information and meaning**. Traditional extractive summarization methods often fail** to produce coherent and concise summaries, as they merely select and concatenate sentences from the original text.

Abstractive summarization, on the other hand, generates new sentences that capture the essence of the source material, making It **more effective** for creating readable and informative summaries.

PEGASUS (Pre-training with Extracted Gap-sentences for Abstractive Summarization) is a state-of-the-art model designed to address the challenges of abstractive summarization. It leverages a novel self-supervised objective during pre-training, where important sentences are removed from the input document, and the model is tasked with generating these sentences from the remaining text. This approach helps the model learn to distill and generate coherent summaries, achieving superior performance on various summarization tasks**.**

Feasibility Analysis

Technical Feasibility:

1. **Model Architecture:** PEGASUS is based on the Transformer encoder-decoder architecture, which is highly effective for sequence-to-sequence tasks like summarization. The model's ability to handle long sequences and model dependencies makes it suitable for summarizing lengthy documents.
2. **Pre-training Objective*:*** The gap-sentence generation objective used in PEGASUS pre-training closely mirrors the summarization task, leading to better fine-tuning performance. This self-supervised approach allows for the creation of large-scale training data without human annotation, which is often a bottleneck in supervised learning.
3. **Performance:** PEGASUS has demonstrated state-of-the-art performance on multiple summarization datasets, including news, science, and legislative documents. Its ability to generate high-quality summaries has been validated through extensive experiments.

Economic Feasibility:

1. **Cost of Implementation:** Implementing PEGASUS requires access to computational resources for training and fine-tuning the model. Cloud-based platforms like Google Cloud and AWS offer scalable solutions for training large models, making it economically feasible for organizations with sufficient budgets.
2. **Return on Investment:** The improved quality of summaries generated by PEGASUS can lead to significant time savings and enhanced productivity for users who need to process large volumes of text. This can be particularly beneficial for industries like media, legal, and academic research, where efficient information extraction is critical.

Operational Feasibility:

1. **Ease of Integration:** PEGASUS can be integrated into existing NLP pipelines and applications using frameworks like TensorFlow and Hugging Face's Transformers library. This makes it accessible for developers and researchers looking to enhance their text summarization capabilities.
2. **Scalability:** The model's architecture and training approach allow it to scale effectively with the availability of more data and computational resources. This ensures that PEGASUS can handle increasing demands for summarization tasks as the volume of text data grows.

5. PEGASUS MODEL

PEGASUS (Pre-training with Extracted Gap-sentences for Abstractive Summarization) is a state-of-the-art model designed to address the challenges of abstractive summarization. It leverages a novel self-supervised objective during pre-training, where important sentences are removed from the input document, and the model is tasked with generating these sentences from the remaining text. This approach helps the model learn to distill and generate coherent summaries, achieving superior performance on various summarization tasks.

In PEGASUS, important sentences are removed/masked from an input document and are generated together as one output sequence from the remaining sentences.

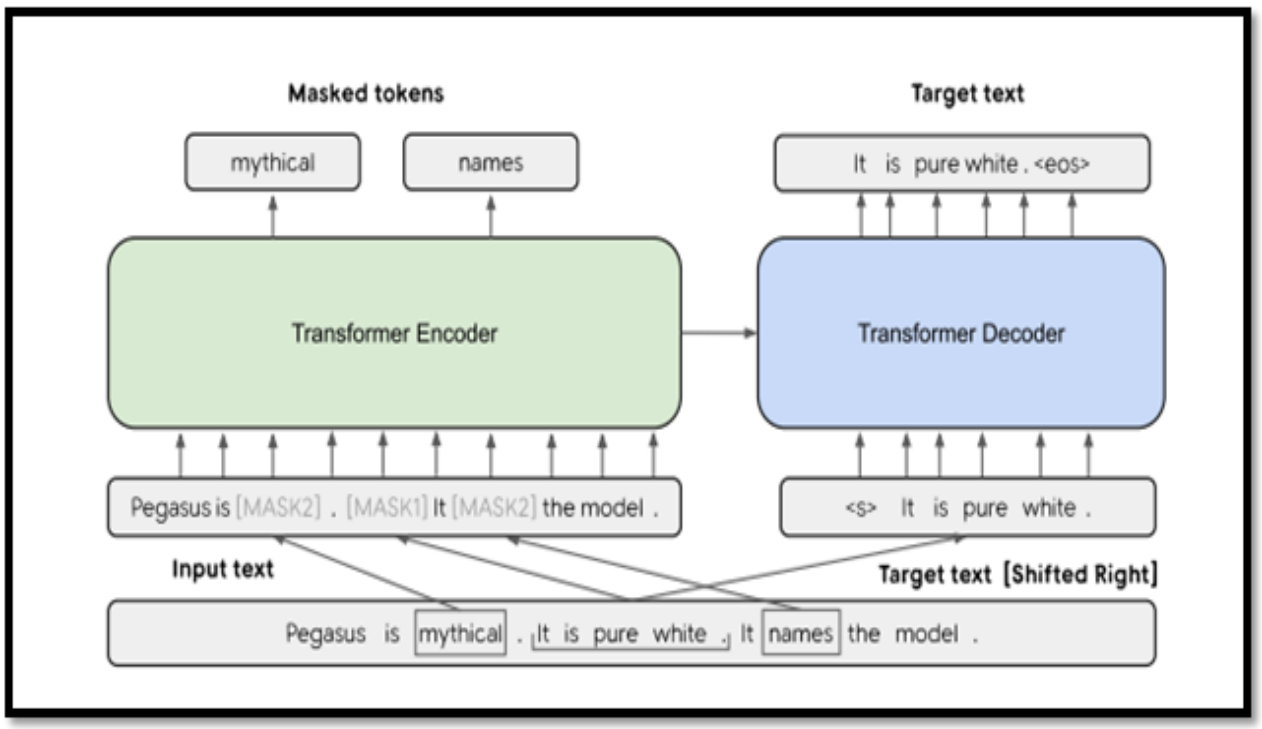

Fig:3-*Base Architecture of Pegasus Model*

6. IMPLEMENTATION FOR TEXT SUMMARIZATION IN PEGASUS

1. Install the Transformer
2. Import the Library
3. Select the GPU
4. Define the model Name
5. Initialize the Tokenizer
6. Initialize the Model
7. Load the Data
8. Fine Tuning the Model
9. Evaluate the Model
10. Save the Model & the Tokenizer
11. Test the Model

1. Install the Transformer:

The first step is to install the transformers library, which provides us with pre-trained models and tokenizers.

```
!pip install transformers[torch]
!pip install accelerate
```

2. Import the Library:

Import the necessary modules to work with the model and data.

```
from transformers import pipeline, set_seed
import matplotlib.pyplot as plt
from datasets import load_dataset
import pandas as pd
from datasets import load_dataset, load_metric
from transformers import AutoModelForSeq2SeqLM, AutoTokenizer
import torch
```

3. Select the GPU:

Use GPU acceleration for faster training and inference.

device = "cuda" if torch.cuda.is_available() else "cpu"

print(f"Using device: {device}")

4. Define the model Name and Initialize the Tokenizer and Model:

- Specify the pre-trained model's name or path. For PEGASUS on CNN/DailyMail data.
- Before processing data, we need a tokenizer for our specific model.
- Load the pre-trained model into memory.

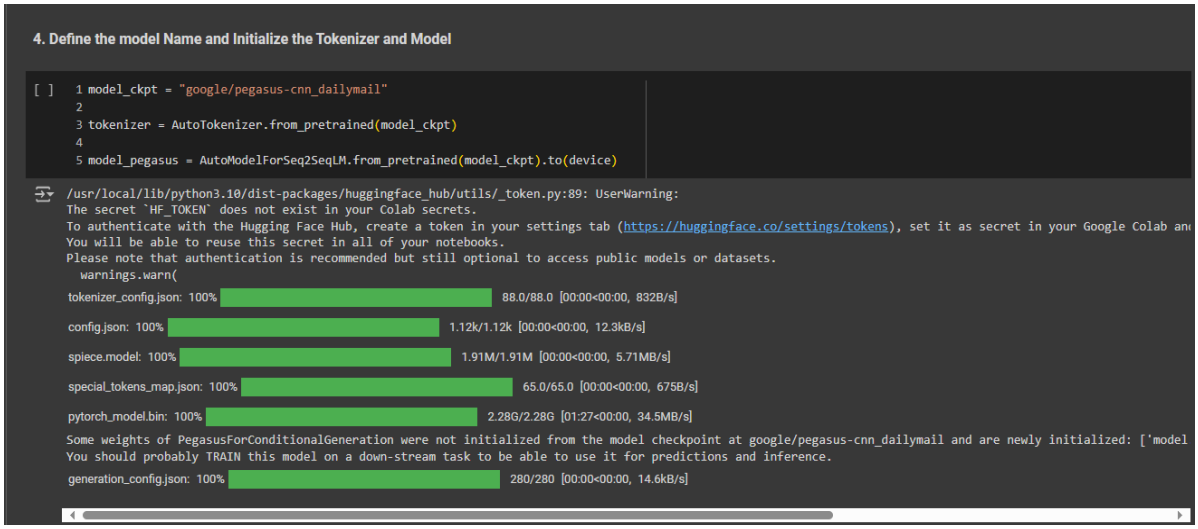

Fig:4-*Defining the CNN/DailyMail data on Pegasus model*

def generate_batch_sized_chunks(list_of_elements, batch_size):

for i in range(0, len(list_of_elements), batch_size):

yield list_of_elements[i : i + batch_size]

calculates a given metric (like ROUGE) t

```python
def calculate_metric_on_test_ds(dataset, metric, model, tokenizer,
                                batch_size=16, device=device,
                                column_text="article",
                                column_summary="highlights"):
# Splitting the dataset into batches
    article_batches = list(generate_batch_sized_chunks(dataset[column_text], batch_size))
    target_batches = list(generate_batch_sized_chunks(dataset[column_summary], batch_size))

# Processing each batch
    for article_batch, target_batch in tqdm(
        zip(article_batches, target_batches), total=len(article_batches)):

# Tokenizing the articles:
        inputs = tokenizer(article_batch, max_length=1024,  truncation=True,
                        padding="max_length", return_tensors="pt")

# Generating summaries:
        summaries = model.generate(input_ids=inputs["input_ids"].to(device),
                           attention_mask=inputs["attention_mask"].to(device),
                           length_penalty=0.8, num_beams=8, max_length=128)
        ''' parameter for length penalty ensures that the model does not generate sequences that are too long. '''

        # Finally, we decode the generated texts,
        # replace the  token, and add the decoded texts with the references to the metric.
        decoded_summaries = [tokenizer.decode(s, skip_special_tokens=True,
                                clean_up_tokenization_spaces=True)
               for s in summaries]

        decoded_summaries = [d.replace("", " ") for d in decoded_summaries]

# Updating the metric
        metric.add_batch(predictions=decoded_summaries, references=target_batch)

    #  Finally compute and return the ROUGE scores.
    score = metric.compute()
    return score
```

Fig:6-*Rouge Score calculation*

1. **<u>generate_batch_sized_chunks</u>, is a utility function designed to split a list into smaller, batch-sized chunks.**

2. **<u>calculate_metric_on_test_ds,</u> calculates a given metric (like ROUGE) on a test dataset using a given model and tokenizer.**

<u>5. Loading the Dataset:-</u>

- **Content:** The dataset contains dialogues between two participants, typically focusing on everyday topics. Each dialogue is paired with a human-written summary.
- **Size:** The SAMSUM dataset includes around 16,000 dialogues, providing a substantial amount of data for training summarization models.
- The SAMSUM dataset is a valuable resource for advancing the field of dialogue summarization, offering both the data and benchmarks needed to improve model performance in generating useful summaries of conversational data.

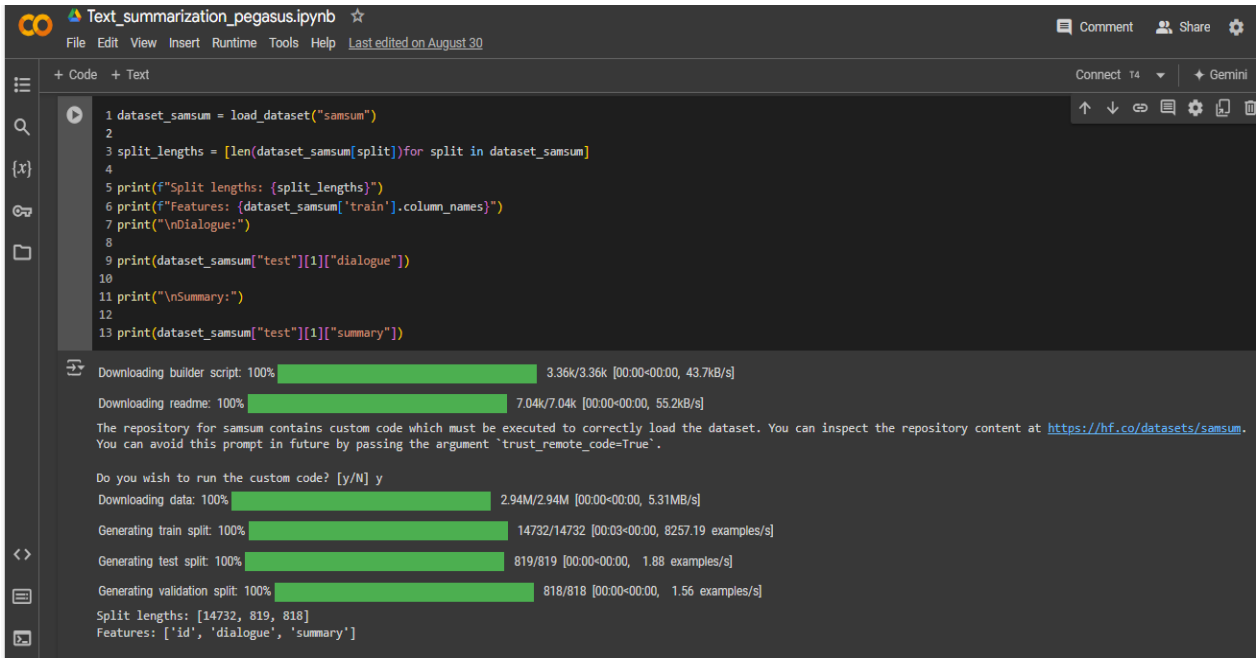

Fig:7-Loading SAMSUM Dataset in Pegasus Model

6. Output:

```
Dialogue:
Eric: MACHINE!
Rob: That's so gr8!
Eric: I know! And shows how Americans see Russian ;)
Rob: And it's really funny!
Eric: I know! I especially like the train part!
Rob: Hahaha! No one talks to the machine like that!
Eric: Is this his only stand-up?
Rob: Idk. I'll check.
Eric: Sure.
Rob: Turns out no! There are some of his stand-ups on youtube.
Eric: Gr8! I'll watch them now!
Rob: Me too!
Eric: MACHINE!
Rob: MACHINE!
Eric: TTYL?
Rob: Sure :)

Summary:
Eric and Rob are going to watch a stand-up on youtube.
```

Fig:8-Text summarization result

7. FUTURE SCOPE

- **Enhanced Summarization Quality:** Future work could focus on improving the quality of summaries generated by Pegasus, especially in terms of coherence, relevance, and informativeness. Fine-tuning models on more diverse datasets and incorporating feedback loops could help address current limitations.

- **Domain-Specific Summarization**: Pegasus has shown strong performance in general summarization tasks. There is potential for developing specialized versions tailored to specific domains like legal, medical, or technical fields, where nuanced understanding is critical.

- **Multilingual Capabilities:** Expanding Pegasus to handle multiple languages with high accuracy can broaden its applicability. Current models often struggle with languages other than English, so creating robust multilingual summarization models is a key area for future research.

- **Real-Time Summarization:** Improving the efficiency of Pegasus to enable real-time summarization could be valuable for applications like live news feeds, real-time translation services, or interactive educational tools.

- **Integration with Other AI Technologies:** Combining Pegasus with other AI technologies, such as question-answering systems or conversational agents, could create more sophisticated tools for users, offering not just summaries but also contextually relevant information and interactive experiences.

8. CONCLUSION

Abstractive text summarization is a rapidly evolving field with significant advancements driven by neural network-based models and pre-trained language models. Despite the progress, challenges such as maintaining coherence and developing better evaluation metrics persist, offering ample opportunities for future research.

The Pegasus model represents a significant advancement in the field of text summarization, leveraging sophisticated natural language processing techniques to produce high-quality, coherent, and contextually relevant summaries. Its success in understanding and condensing complex information demonstrates its potential to revolutionize how we interact with and process textual data.

Looking ahead, there are several exciting opportunities to further enhance the capabilities of Pegasus. By focusing on improving summarization quality, expanding domain-specific applications, developing multilingual models, and integrating with other AI technologies, Pegasus can become even more powerful and versatile. Additionally, addressing ethical concerns, personalizing summaries, and optimizing energy efficiency will be crucial in ensuring that this technology is used responsibly and effectively.

9. REFERENCES

1. "**Abstractive text summarization: state of the art, challenges, and improvements"** by *hassanshakil, ahmadfarooq, and jugal kalita*. this paper provides a comprehensive overview of abstractive text summarization techniques, challenges, and future research directions.

2. "**A Systematic Survey of Text Summarization: From Statistical Methods to Large Language Models**" by *Haopeng Zhang, Philip S. Yu, and Jiawei Zhang*. This survey covers the evolution of text summarization methods, from traditional statistical approaches to modern large language models.

3. "**Text Summarization Techniques: A Brief Survey**". This paper reviews various automatic text summarization methods, highlighting their effectiveness and limitations.

4. "**Attention Is All You Need**" by *Ashish Vaswani et al*. This paper was presented at the 2017 Conference on Neural Information Processing Systems (NeurIPS). It introduced the Transformer model, which relies entirely on self-attention mechanisms, dispensing with recurrence and convolutions.

5. "**HuggingFace's Transformers: State-of-the-art Natural Language Processing"** by *Thomas Wolf et al.,* presented at the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). This paper discusses the development and impact of the HuggingFace Transformers library.

10. BIO DATA

Name: U Suriya

Contact No: 9566449325

Email ID: usuriya0809@gmail.com

D.O.B: 08-09-2003

Gender: Male

Nationality: Indian

Languages knows: Tamil & English

Skills: python & SQL

Education Details:

S.NO	Education	College/School	Year of pass	Pass percentage
1	B.Tech - IT	Anjalai Ammal Mahalingam Engineering College	2025	
2	12thClass	Trinity Academy CBSE School	2021	78%
3	10thClass	Trinity Academy CBSE School	2019	75%

Projects:

S.NO	Domain	Project Title	University/School
1	JAVAFX	College Transportation Application	Anjalai Ammal Mahalingam Engineering college
2	DBMS	Smart Banking	Anjalai Ammal Mahalingam Engineering college
3	Software Engineering	SDLC Architecture – Recruitment System	Anjalai Ammal Mahalingam Engineering college

Experience: Fresher

Address: 597/A, Amman Nagar,
Main Road, Neelapadi,
Nagapattinam – 611105

Declaration: I hereby declare that all the above information is true and correct to the best of my knowledge.

Place: Thiruvarur

Date: **Signature**