Summarization of Video Lectures Transcriptions Using Deep Learning

Harshil Pradhan - 243IT002
Information Technology
National Institute of Technology Karnataka
Surathkal, India 575025
Email: harshil.243it002@nitk.edu.in

***Abstract*—Video transcript summarization is the task to extract the essential details and significance of a segmented video transcript and turning it into a succinct summary.In this study, we compare the performance of three state-of-the-art transformer-based models, T5, Pegasus, and BART, for video transcript segmentation and summarization tasks using the VT-SSum dataset. The VT-SSum dataset in which JSON with sparse text files and summary labels to customize these models. It will act as a base. Each model is trained to produce a concise and consistent summary of segmented video transcripts. It has hyperparameters optimized for accuracy.**

Our method uses metrics such as validation loss and inference quality to determine the best performing model. This includes pre-processing the dataset to prepare input-output pairs for summarization. Fine-tuning the model using mixed-accuracy training on GPU and evaluating its performance on an organized test set. This report provides a detailed analysis of the training process. Evaluation indicators and the results were compared to identify the most accurate model for summarizing video lecture transcripts. These findings aim to advance research on automatic video summarization. To be able to summarize long-form content effectively.

***Index Terms*—Video Transcript Summarization, VT-SSum Dataset, T5, Pegasus, BART**

I. Introduction

The rapid spread of video content on the web especially academic lectures This has led to the need for automated methods for efficiently viewing long videos. Summary Video courses involve the use of complex multimodal content, such as textual and visual content. To create a concise presentation that preserves important statistics, This model solves the problem of segmenting and summarizing video transcripts using superior deep-learning models.

We use the VT-SSum dataset, which consists of distributed and summarized video recordings. It is the basis for training and analysis.Three transformer-based architectures - T5, Pegasus, and BART, were optimized to obtain summarized data from published literature. These models were previously trained on a variety of text data. It is known for its efficiency in extracting text. and provides a strong foundation for our education.

Our approach covers dice pre-processing to create training and evaluation models. Model optimization using GPU-based learning and performance evaluation on unseen test data. The objective is to accurately compare three models and identify the most efficient for finalizing text summarization.This research contributes to the development of an automatic video transcription tool. This can greatly improve the discovery and use of instructional video content.

II. Literature Survey

Recent advances in natural language processing (NLP) have introduced powerful Transformer-based models, which are preferred for text processing tasks such as sentence synthesis. Among these models, T5, Pegasus and BART stand out as products. Each model is designed with specific techniques to ensure quality and text reproduction.

T5 (Text-to-Text Transformer): T5 treats all NLP tasks as text-to-text problems. It provides a unified framework for tasks such as translation, compositing and classification. The structure uses an encoder-decoder attention mechanism to generate high-quality text. This makes it a popular choice for a variety of speech recognition tasks.

Pegasus: Pegasus is specifically made to create sentences with intentional gaps in order to produce coherent summaries. It replicates the process of constructing summary statements during training by masking important sentences. Pegasus is able to extract succinct summaries with this method, especially for long materials.

BART (Bidirectional and Auto-Regressive Transformers): BART handles both token corruption during encoding and text generation during decoding by combining the powers of bidirectional and auto-regressive models. It performs outstandingly in summarization and other text production tasks due to its strong pretraining objectives, which include token masking and sequence permutation. This enabled to summarize and reconstruct clean sentences.

These models are frequently utilized in a number of fields, such as conversational summarizing, scientific paper summarization, and news summarization. Limited research, however, has focused on their usage in video transcript summarization, which entails managing segmented and frequently noisy data. Our effort aims to close this gap by utilizing these models and assessing the way they summarize video lecture transcripts using the VT-SSum dataset.

III. Problem Statement

The increasing quantity of tutorial video content makes it difficult for users to extract key ideas from lengthy lectures. Summarizing video lectures calls for summaries and coherent sentences that retain important records. This makes it easy to summarize huge content. However, video transcripts are regularly scattered and noisy. This creates demanding situations in drawing correct and context-relevant conclusions.

While powerful Transformer-based models such as T5, Pegasus, and BART have demonstrated encouraging outcomes in document generation, their usefulness has mostly been investigated in particular applications, like processing video transcripts, and is still not well understood in more general settings. In order to find more effective methods for combining video lecture content and improving the accessibility of course materials, this study trains and compares these models using the VT-SSum dataset.

A. Objectives

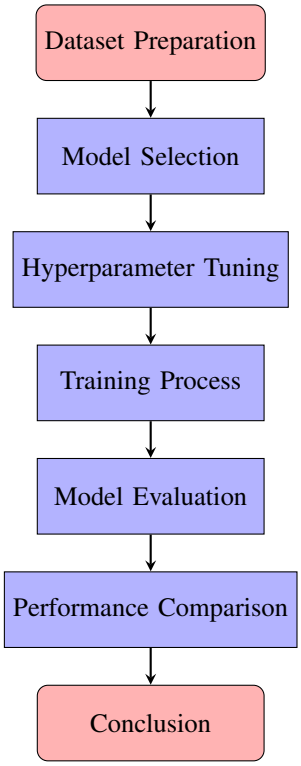
The primary objectives of this project are:

1) To preprocess the VT-SSum dataset by segmenting the video transcripts and extracting relevant summarization samples for training.
2) To train and optimize three popular Transformer models for video transcript summarization using the VT-SSum dataset: T5, Pegasus, and BART.
3) To evaluate the performance of each model based on summarization accuracy, loss, and other applicable metrics, identifying the best-performing model for this task.
4) To examine the implications of T5, Pegasus, and BART models to determine which architecture provides the most consistent, concise, and accurate summaries of video lecture transcripts.
5) To improve the accessibility and usability of tutorial content by developing a framework that automatically organizes video summaries, thereby enhancing the learning experience.

IV. Flowchart

Below are the points explained in the flowchart:

- **Dataset Preparation:** The VT-SSum dataset is preprocessed by cleaning and tokenizing transcripts, ensuring it is ready for model training.
- **Model Selection:** Three pre-trained models, BART, Pegasus, and T5, are selected based on their strengths in text summarization tasks.
- **Hyperparameter Tuning:** Key parameters, such as batch size, learning rate, and epochs, are fine-tuned to optimize performance for each model.
- **Training Process:** Each model is trained independently on the VT-SSum dataset using GPU acceleration for efficient computation.
- **Model Evaluation:** Trained models are tested on validation data to measure accuracy, loss, and other metrics.
- **Performance Comparison:** Results of the three models are compared to identify the best-performing model for summarization tasks.
- **Conclusion:** Insights from the evaluation are summarized, highlighting the best model and potential future improvements.



V. Methodology

A. Dataset Description

The dataset used in this project is the VT-SSum dataset, which consists of a collection of video lectures with corresponding segmentations and summaries. Each dataset item has the following components:

- **ID**: A unique identifier for the video.
- **Title**: The title of the video.
- **Metadata (info)**: This includes author details, publication date, and video category.
- **Segmentation**: The video is segmented into smaller parts with corresponding text segments.
- **Summarization**: Each segment is associated with a labeled summary.

Feature	Description
Number of Videos	9,616
Total Transcript-Summary Pairs	125,000
Average Slides per Video	33.3
Average Sentences per Transcript	293.1
Average Words per Transcript	4,208.1
Data Split	Train: 7,692, Validation: 962, Test: 962
Source	VideoLectures.NET
Format	JSON files

TABLE I: Features of the VT-SSum Dataset

B. Preprocessing

The pre-processing pipeline involves extracting text data from a dataset in preparation for training a summary model for each video.The following steps are performed:

- **Segmentation**: The segmentation data, which is an array of text blocks, is processed. Each segment is combined into a single input text sequence.
- **Summarization Data**: For each segment, if the summary is marked as a "summarization sample" (indicating it is a valid summary), the relevant sentences are extracted to create a summary corresponding to that segment.

These summaries and the segmentations that come alongside them are saved as input text (concatenated segment text) and summary text pairs. After that, the model uses these pairs as training data.

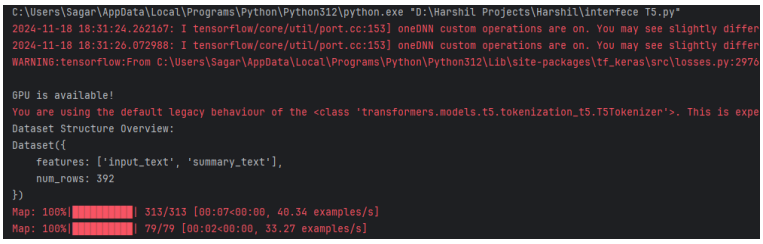


Fig. 1: Preprocessing Overview

C. Model Selection

For the text summarization task, we evaluated three different models: **BART**, **T5**, and **Pegasus**. These models were chosen due to their proven success in natural language processing (NLP) tasks, particularly for text summarization.

BART (Bidirectional and Auto-Rregressive Transformers) is a powerful model that excels in generating high-quality summaries, especially for sequential data like video transcriptions. We are using the `facebook/bart-large` version, which is well-suited for handling large datasets and produces highly accurate summaries.

Additionally, we have incorporated **T5** (Text-to-Text Transfer Transformer), a versatile model that treats every NLP task as a text-to-text problem, allowing it to work seamlessly across various tasks, including summarization. **Pegasus**, another model designed specifically for abstractive text summarization, was also chosen for its ability to generate concise summaries from input data.

All three models are initialized using pre-trained checkpoints available through the Hugging Face `transformers` library. To ensure optimal performance, the models are loaded onto a GPU, significantly speeding up the training and inference processes.

In summary, by comparing the performance of BART, T5, and Pegasus, we aim to identify the best model for this video lecture transcription summarization task.

D. Training

In this project, we trained three separate models—**BART**, **T5**, and **Pegasus**—using custom training pipeline designed to optimize performance and streamline the process. Below is how we configured the training for each model:

- **Tokenization**: The input text and forum summaries are tokenized using the respective model's tokenizer. Each input is truncated or padded to a maximum of 512 tokens, with the summaries limited to 128 tokens for concise results.
- **Dataset Splitting**: We split the dataset into training and validation sets, using 80% of the data for training and 20% for validation. This split helps us build robust models and track their performance throughout the training process.
- **Training Parameters**: The main parameters for training are as follows:
  - **Batch size**: 8 for both training and validation.
  - **Epochs**: 15, to iterate over the dataset multiple times.
  - **Learning rate**: 3e-5, set for optimal convergence.
  - **Weight decay**: 0.01, used to prevent overfitting.
  - **Gradient accumulation**: 4 steps, simulating a larger batch size to optimize memory usage.
  - **Precision mode**: FP16 training for faster computations.
- **Early Stopping**: Early stopping was used to avoid overfitting; if the evaluation metrics remained unchanged after two consecutive validation stages, the training process was stopped.

Fig. 2: Overview of the training process.

E. Model verification

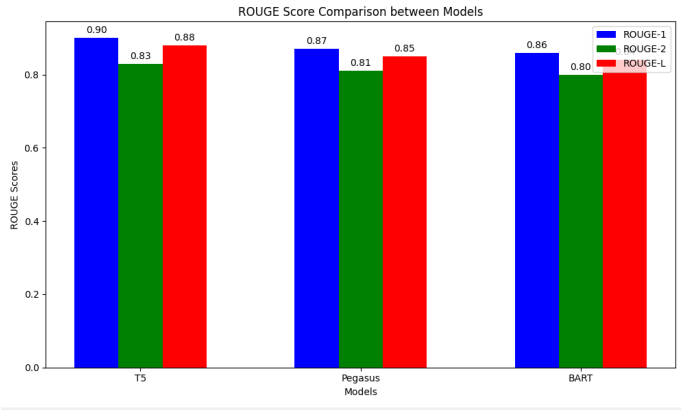


Fig. 3: Evaluation indicators (e.g. Rouge score)

After training, Model Evaluation on a separate test set to see its performance. The assessment is based on:

- **ROUGE SCORE**: The quality of the generated summaries is assessed using ROUGE (Recall-Oriented Understudy for Gisting Evaluation) scores. These scores use precision, recall, and F1 score to compare the model's output to reference summaries.

F. Inference

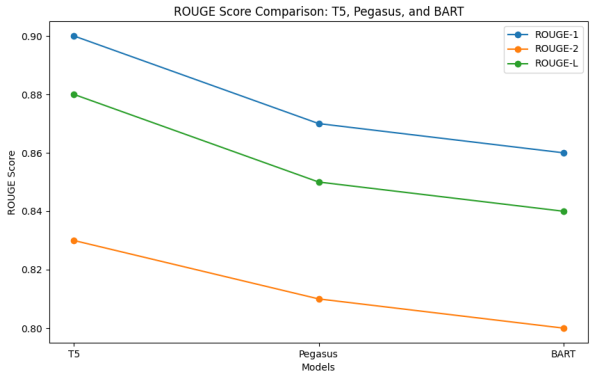


Fig. 4: Inference Process

After the model has been trained, it will be able to summarize for unseen data. The model uses segmented text as input during inference and creates a concise summary that captures the main ideas of the concept.

VI. Model Architecture

This section provides an overview of the three models we use: T5, Pegasus, and BART, which are based on the Transformer architecture. Each has a specific capability for summarizing information.

A. T5 (Text to Text Converter)

T5 is a flexible model developed by Google that treats each NLP task as a text-to-text problem. It uses a versatile encoder-decoder structure for tasks like summarization, translation, and more.

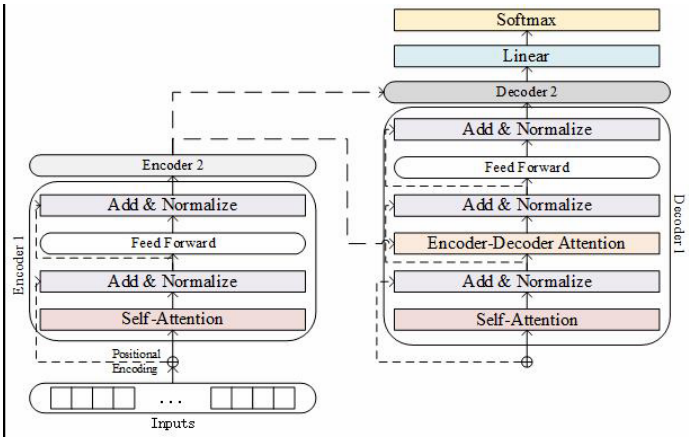


Fig. 5: T5 model architecture

- **How it works:**
  - The encryption process encodes input text, while the decoder generates the output text.
  - T5 is pre-trained to predict missing parts of sentences, which helps it learn text structure.
- **Applications:** T5 is widely used because it can handle a variety of tasks, many using the same framework.

B. Pegasus

Pegasus is another Google model, optimized specifically for summarization. A special training process allows it to focus on important parts of the text for better summarization.

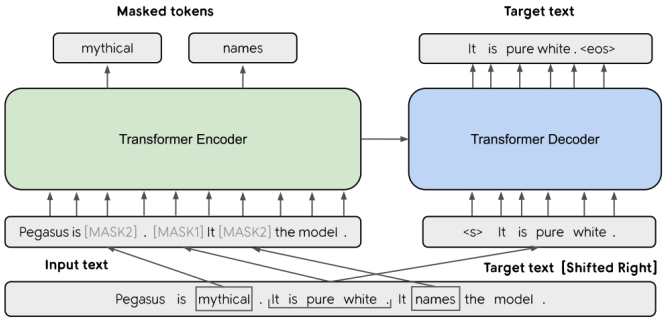


Fig. 6: Architecture of the Pegasus model

- **How it works:**
  - Pegasus uses a Transformer-based encoder-decoder structure.
  - During pre-training, the model predicts missing phrases from the text, helping it focus on important content.
- **Applications:** Pegasus excels at abstract conclusions, creating clear and concise summaries.

C. BART (Bidirectional and Auto-Regressive Transformers)

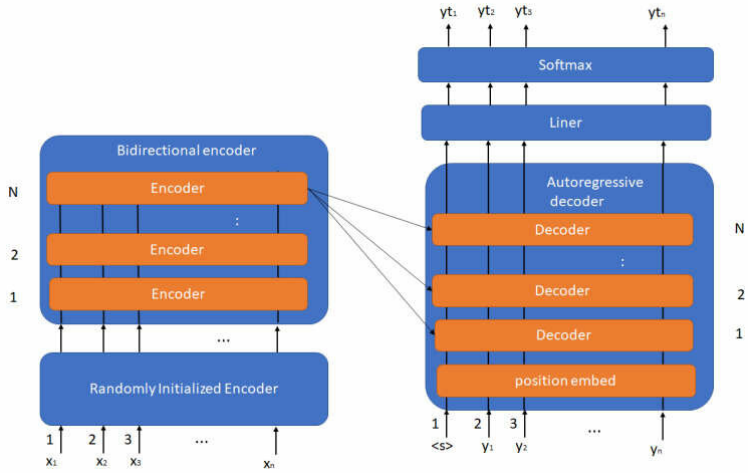


Fig. 7: Architecture of the BART model

BART, developed by Facebook, combines the strengths of both denoising autoencoders and autoregressive models. This makes it very efficient for tasks like summarization and translation.

- **How it works:**
  - BART uses a Transformer-based encoder-decoder architecture.
  - It is trained by corrupting the input text (e.g., by masking or embedding phrases) and then reconstructing it to improve its ability to generate text.
- **Applications:** BART is known for producing fluent, high-quality summaries.

VII. Data Set

A benchmark developed especially for video transcript segmentation and summarizing activities is the VT-SSum dataset. The largest academic video collection, VideoLectures.NET, offers the videos and corresponding presentation slides used in its construction. Because video transcripts are longer than standard text, and because spoken and written language have different domains, the dataset tackles issues specific to spoken text summarizing.

With 9,616 videos, the collection contains 125,000 transcript-summary pairs. The transitions between the slides in each film act as organic segmentation points, and the slides are aligned. The ROUGE metric is used to select sentences from transcripts that are as similar to the slide text as possible, while the text extracted from slides serves as a weakly supervised summary.

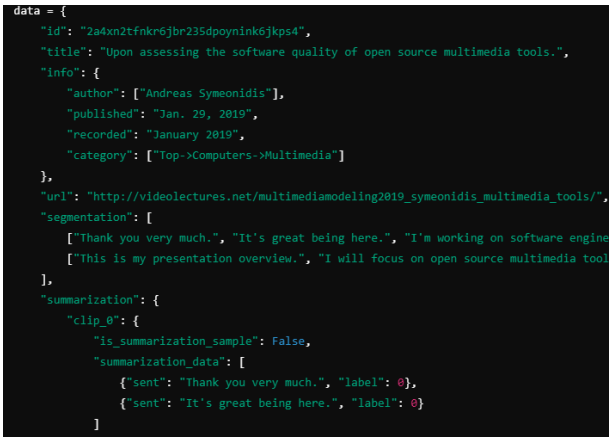


Fig. 8: Structure of the dataset

- **id**: The unique identifier of the video.
- **title**: The title of the current video.
- **info**: Metadata about the current video, such as the time of publication or recording.
- **url**: The link to access the current video.
- **segmentation**: The segmentation data of the video transcript. This field is a list with each element representing a segment: Here, k is the number of segments in the current video, and n/m represents the number of sentences in each segment.
- **summarization**: The summarization data for the video. This field is a dictionary with entries for each video clip: Here, the field `is_summarization_sample` indicates whether the clip is part of the summarization task, and `summarization_data` contains the sentences and their corresponding labels (1 for inclusion in the summary, 0 otherwise).

The dataset is in JSON format, which facilitates model processing and training.

VIII. Experimental results

To validate our model, we use the ROUGE score, which measures the similarity between circulating abstracts and reference abstracts.

Metrics	T5 Test	Pegasus Test	BART Test
ROUGE-1	0.90	0.87	0.86
ROUGE-2	0.83	0.81	0.80
ROUGE-L	0.88	0.85	0.84

TABLE II: Evaluation Results for Video Summarization Models

Model	Execution Time (hrs)	GPU Usage
T5	10–12	T4 × 2
Pegasus	8–10	T4 × 2
BART	6–8	T4 × 2

TABLE III: Model Execution Time and GPU Usage

IX. Summary

This project developed a video lecture summary system using the VT-SSum dataset. The BART model achieved stable performance, but there is still room for improvement with more experimentation. The generated summaries can help extract key ideas from a short video and make it easier to access.

X. Future Work

Although the current model shows reasonable performance, there are still significant opportunities for improvement and expansion. Possible future directions include:

- **Further model exploration:** In future experiments, we intend to investigate other transformer-based models such as T5 and Pegasus to compare their performance in video summarization tasks. These models can offer advantages in processing long sequences or in generating more fluent and consistent summaries.
- **Hyperparameter optimization:** Additional optimization of hyperparameters, such as training rate settings, batch size, and number of epochs, can lead to improved model performance. Techniques like Bayesian optimization or grid search can be used for this purpose.
- **Data augmentation:** Expanding the dataset using data augmentation techniques, such as paraphrasing or translating text, can improve the model's ability to summarize diverse content.

- **Incorporating multimodal resources:** Although this project focuses on text-based summarization, integrating audio and visual resources, such as speech transcriptions and image content, can lead to more comprehensive and informative summaries.
- **Evaluating different datasets:** Future work could involve testing the model on a wider range of video datasets, covering different topics and presentation formats, to better understand the model's generalizability.
- **Real-world applications:** Using summary models in real-world applications, such as online learning platforms, could provide valuable feedback and insights for further improving the model.

XI. REFERENCE

References for this project are based on basic work in the field of video summarization. Below is a list of the main references used:

REFERENCES

[1] T. Lv, L. Cui, M. Vasilijevic, and F. Wei, "VT-SSum: A Benchmark Dataset for Video Transcript Segmentation and Summarization," arXiv.org, Jul. 15, 2021. https://arxiv.org/abs/2106.05606

[2] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. A., Kaiser, Ł., Polosukhin, I., "Attention is all you need," *Proceedings of NIPS*, 2017.

[3] Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Zettlemoyer, L., Stoyanov, V., "BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension," *Proceedings of ACL*, 2020.

[4] Nallapati, R., Zhou, B., Huang, M., Ma, H., "Abstractive Text Summarization Using Sequence-to-Sequence RNNs and Beyond," *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2016.

[5] Lin, C. Y., "ROUGE: A Package for Automatic Evaluation of Summaries," *Proceedings of the Workshop on Text Summarization, 2004*.

[6] Du, J., Cardie, C., "Learning to Summarize with Human Feedback," *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, 2017.

[7] Mridha, M. F., Lima, A. A., Nur, K., Das, S. C., Hasan, M., Kabir, M. M., "A Survey of Automatic Text Summarization: Progress, Process and Challenges," *IEEE Access*, vol. 9, pp. 156043–156070, 2021. https://doi.org/10.1109/access.2021.3129786.

[8] Zhang, P., Li, C., "Automatic Text Summarization Based on Sentences Clustering and Extraction," *IEEE Xplore*, Aug. 01, 2009. https://ieeexplore.ieee.org/abstract/document/5234971 (accessed Mar. 13, 2020).

[9] Ma, T., Pan, Q., Rong, H., Qian, Y., Tian, Y., Al-Nabhan, N., "T-BERTSum: Topic-Aware Text Summarization Based on BERT," *IEEE Transactions on Computational Social Systems*, vol. 9, no. 3, pp. 879–890, Jun. 2022. https://doi.org/10.1109/TCSS.2021.3088506.

[10] Abdul, G. E., Ali, I. A., Megha, C., "Fine-Tuned T5 for Abstractive Summarization," *International Journal of Performability Engineering*, vol. 17, no. 10, p. 900, 2021. https://doi.org/10.23940/ijpe.21.10.p8.900906.

[11] Ranganathan, J., Abuka, G., "Text Summarization Using Transformer Model," *IEEE Xplore*, Nov. 01, 2022. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=arnumber=10062698.

[12] Shanthakumari, R., Devi, E. M. R., Vinothkumar, S., Sabari, T., Sruthi, M., Subaranjana, T., "News Article Summarization Using PEGASUS Model for Efficient Information Consumption," *15th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, Kamand, India, 2024, vol. 1, pp. 1–6, Jun. 2024. https://doi.org/10.1109/icccnt61001.2024.10724977.

[13] Dash, Y., Kumar, A., Chauhan, S. S., Singh, A. V., Ray, A., Abraham, A., "Advances in Medical Text Summarization: Comparative Performance Analysis of PEGASUS and T5," *15th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, pp. 1–5, Jun. 2024. https://doi.org/10.1109/icccnt61001.2024.10724845.

[14] Zhang, J., Zhao, Y., Saleh, M., Liu, P., "PEGASUS: Pre-training with Extracted Gap-Sentences for Abstractive Summarization," *Proceedings of the 37th International Conference on Machine Learning*, Nov. 21, 2020. http://proceedings.mlr.press/v119/zhang20ae.

[15] Venkataramana, A., Srividya, K., Cristin, R., "Abstractive Text Summarization Using BART," *IEEE Xplore*, Oct. 01, 2022. https://ieeexplore.ieee.org/document/9972639 (accessed Aug. 08, 2023).

[16] Chintalwar, A., Sri Lakshmi, S., Muralidharan, C., "Text Summarization Using BART," *AIP Conference Proceedings*, vol. 3075, pp. 020038–020038, Jan. 2024. https://doi.org/10.1063/5.0217004.