YOUTUBE VIDEO TRANSCRIPT SUMMARIZATION USING GEMINI-PRO AND PEGASUS

A PROJECT REPORT

submitted by

EVELIN MANOJ

(TCR22CSCE07)

to

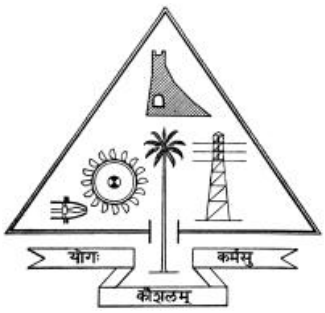
the APJ Abdul Kalam Technological University in partial fulfillment of the requirements for the award of the Degree

of

Master of Technology

in

Computer Science and Engineering



Department of Computer Science and Engineering

Government Engineering College

Thrissur

JUNE 2024

DECLARATION

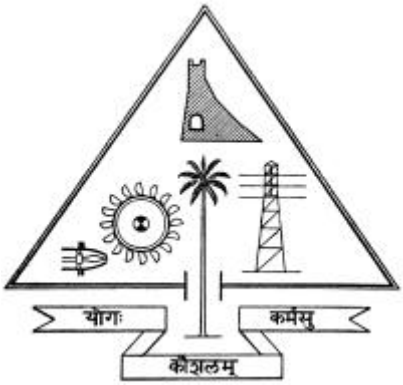
I undersigned hereby declare that the project report **"YOUTUBE VIDEO TRANSCRIPT SUMMARIZATION USING GEMINI-PRO AND PEGASUS"**, submitted for partial fulfillment of the requirements for the award of the degree of Master of Technology of the APJ Abdul Kalam Technological University, Kerala is a bonafide work done by me under supervision of **Prof. Rahmathulla K**, Assistant Professor Department of CSE. This submission represents my ideas in my own words and where ideas or words of others have been included, I have adequately and accurately cited and referenced the original sources. I also declare that I have adhered to the ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

Place: Signature

Date: EVELIN MANOJ

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

GOVERNMENT ENGINEERING COLLEGE,THRISSUR



CERTIFICATE

This is to certify that the report entitled **"YOUTUBE VIDEO TRANSCRIPT SUMMARIZATION USING GEMINI-PRO AND PEGASUS"**, submitted by **EVELIN MANOJ** to the APJ Abdul Kalam Technological University in partial fulfillment of the requirements for the award of the Degree of Master of Technology in Computer Science and Engineering is a bonafide record of the project carried out by her under my guidance and supervision. This report in any form has not been submitted to any other University or Institute for any purpose.

Internal Supervisor(s) **External Supervisor(s)**

Project Coordinator **Head of the Department**

ACKNOWLEDGEMENT

I honor and thank the Lord, who has given me insight and made it possible for me to successfully finish this project.

I would like to thank the Department of Computer Science and Engineering, Government Engineering College, Thrissur, for giving me the opportunity to present this project.

I would also like to thank **Dr. Ajay James**, HOD and project coordinator, Department of Computer Science and Engineering for rendering his help and guidance during the completion of this project.

I would like to express my deep gratitude to our PG Co-ordinator **Dr. Shibly Joseph** for rendering her help and guidance throughout this project.

I would also like to thank my guide **Prof. Rahmathulla K** for his timely suggestions encouragement and guidance through the processes involved in the presentation of this project and the preparation of the report.

Lastly, I am grateful to my parents and friends from the bottom of my heart for their inspiration and support throughout this project.

ABSTRACT

Imagine an app that saves you hours by summarizing YouTube video transcripts, so you get all the key information without the fluff. With countless videos uploaded every day, it's often a challenge to find exactly what you need. Creators frequently use clickbait titles and spend excessive time promoting products, making it hard to locate the valuable content within their videos. This innovative app cuts through the noise by providing concise summaries of video transcripts, ensuring you find what you're looking for quickly and effortlessly.

This solution employs cutting-edge AI technology, specifically the Gemini-pro and fine-tuned Pegasus models, to expertly condense transcripts while preserving their original meaning. Here's how it works: simply input a video link into the app, and it utilizes the youtubetranscriptApi in Python to fetch the transcript. The transcript summarization module then processes the text, capturing intricate patterns and dependencies within the content to create a precise and concise summary.

But that's not all. To make the information even more accessible, the app also converts the text summary into an audio format using Google Text-to-Speech (GTTS). This dual-format delivery allows users to either read or listen to the summarized content, catering to different preferences and on-the-go needs.

Imagine the convenience of getting straight to the core information of any lengthy video without wasting time on irrelevant details. Whether you're a student, professional, or just an avid learner, this app enhances your YouTube experience, making it easier than ever to access the valuable information you seek. Save time, avoid frustration, and get to the heart of the content with our revolutionary YouTube transcript summarization app.

Keywords: *Transcripts, text summarization, Gemini-pro, finetune, Pegasus, GTTS.*

CONTENTS

List of Tables

List of Figures

List of Abbreviations

AI	Artificial Intelligence
E2E SSum	End-to-end speech summarization
GCN	Graph Convolutional Networks
GPaS	Graph-based partition-and-summarization
GTTS	Google Text–to–Speech
LSTM	Long Short-Term Memory
ML	MAchine Learning
NLTK	Natural Language Toolkit
NLP	Natural Language Processing
RNN	Recurrent Neural Network
SVM	Support Vector Machines
TTS	Text–to–speech

CHAPTER 1

INTRODUCTION

Text summarization serves as a valuable tool in our information-driven era, allowing us to distill extensive written content into a concise form while retaining its essence. This process is particularly crucial in the context of vast digital content, where the ability to quickly and efficiently extract pertinent information becomes paramount. Automatic text summarization, as a technological solution, offers the means to navigate through extensive textual data with minimal effort, enabling users to access relevant content promptly.

Text summarization can be broadly categorized into two main types: extractive summarization and abstractive summarization. Each approach has its own techniques and applications, offering distinct advantages and challenges[2].

Extractive text summarization involves selecting key sentences, phrases, or sections directly from the source text and concatenating them to create a summary[3]. This method relies on identifying and extracting existing sentences based on their relevance and importance. Techniques used in extractive summarization include sentence scoring, where sentences are evaluated based on features like term frequency, sentence position, and length. Ranking algorithms such as TF-IDF, LexRank, and TextRank are also employed to rank sentences by importance. Additionally, machine learning models like support vector machines (SVM) and neural networks can be trained to recognize significant sentences. The main advantages of extractive summarization are its simplicity and the preservation of the original meaning since the text is not altered. However, it can lead to coherence issues, as the extracted

sentences may not flow well together, and there might be redundancy if the key sentences are not well-selected.

Abstractive text summarization generates new sentences that convey the main ideas of the source text, paraphrasing and rephrasing content to create a summary. This approach often requires a deeper understanding of the text. Techniques used in abstractive summarization include neural networks, specifically recurrent neural networks (RNNs), Long Short-Term Memory (LSTM) networks, and Transformer models like BERT, GPT, and T5. Sequence-to-sequence (Seq2Seq) models with attention mechanisms are also commonly used to convert the input text into a concise summary. Pre-trained language models such as BART, T5, or GPT-3 can be fine-tuned for the specific task of summarization. The advantages of abstractive summarization include producing more natural and coherent summaries, which are often closer to human-generated summaries, and providing contextually relevant information. However, abstractive summarization is more complex to implement, requires extensive computational resources and training data, and there is a risk of distorting the original meaning or omitting critical details.

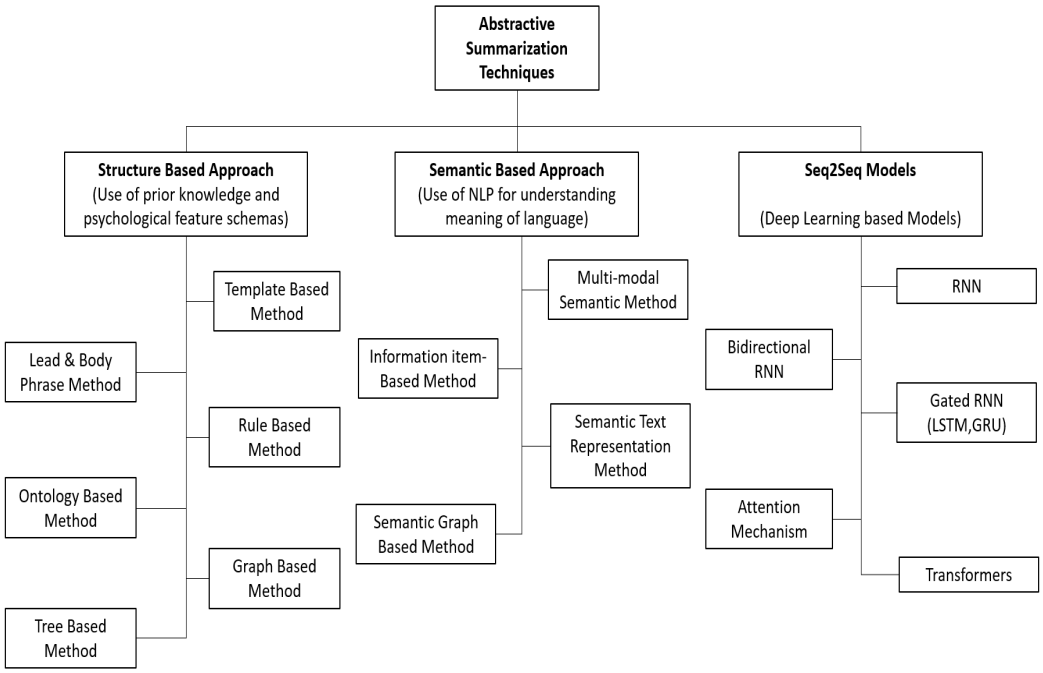


Figure 1.1: Various abstractive text summarization techniques.[1]

Hybrid approaches combine both extractive and abstractive methods to leverage the strengths of each. For instance, an extractive method might first identify key sentences, which are then refined or rephrased using an abstractive model. Techniques for hybrid approaches include two-stage models, where the first stage uses extractive techniques to identify important segments, and the second stage employs abstractive methods to refine and generate the final summary. Reinforcement learning can also be used to balance extraction and abstraction for optimal summaries. Hybrid approaches offer a balanced solution, combining the precision of extractive summarization with the coherence of abstractive summarization, often resulting in higher quality summaries by addressing the weaknesses of purely extractive or abstractive methods[4].

Hybrid approaches combine both extractive and abstractive methods to leverage the strengths of each. For instance, an extractive method might first identify key sentences, which are then refined or rephrased using an abstractive model. Techniques for hybrid approaches include two-stage models, where the first stage uses extractive techniques to identify important segments, and the second stage employs abstractive methods to refine and generate the final summary. Reinforcement learning can also be used to balance extraction and abstraction for optimal summaries. Hybrid approaches offer a balanced solution, combining the precision of extractive summarization with the coherence of abstractive summarization, often resulting in higher-quality summaries by addressing the weaknesses of purely extractive or abstractive methods.

Recognizing the need for efficient information retrieval in the realm of online videos, this proposal centers around the development of an application dedicated to summarizing YouTube video transcripts[5]. As the volume of video content on the internet continues to surge, users often find themselves grappling with the challenge of sifting through numerous videos to locate specific and meaningful information. Compounded by elements such as misleading thumbnails, descriptions, and promotional content, the task of finding accurate and valuable information within a video becomes a time-consuming endeavor[6].

The envisioned application seeks to alleviate this challenge by incorporating a

Transcript Summarization module. The primary objective is to streamline the video-watching experience by providing users with concise yet comprehensive summaries of video transcripts. This approach not only saves time for users but also addresses common frustrations associated with misleading content and prolonged promotional segments within videos.

In essence, this proposed application not only aligns with the growing demand for streamlined information consumption but also addresses the challenges posed by the ever-expanding landscape of online video content. Through automated text summarization, it aims to enhance the user experience by minimizing the time and effort required to access valuable information within YouTube videos.

CHAPTER 2

RELATED WORKS

A review of the literature was done in the project's field. The goal of a literature review is to comprehend the current research and discussions around the subject, and it provides evidence to support practical issues in the field. A literature review contributes to the advancement of knowledge in the topic. I'm going to give a thorough explanation of the literature survey articles here.

2.1 Video To Text Summarization

In 2020, Zhang et al[7]. proposed a graph-based partition-and-summarization (GPaS) framework for dense video captioning, which aims to detect and describe all events in a long untrimmed video. The framework introduces a partition module that segments video proposals, generating visual and textual features for each segment. Further, a summarization module utilizes graph convolutional networks (GCNs) and long short-term memory (LSTM) networks, demonstrating innovative coupling through two interaction modules. The paper validates its effectiveness on the ActivityNet Captions and YouCook II datasets, surpassing state-of-the-art methods. Noteworthy are the paper's clarity, organization, and robust theoretical foundation, substantiated by thorough experimental results, ablation studies, qualitative analysis, and insightful discussions on design choices and limitations.

While the paper excels in its contributions, a more detailed exposition on the joint optimization of GCN and LSTM in an end-to-end manner is warranted. Ad-

ditional insights into training procedures, including loss functions, optimization algorithms, and hyperparameters, would enhance comprehension. The absence of a comparative analysis with recent works utilizing GCN and LSTM for dense video captioning is another noteworthy point. A comparative assessment would provide valuable context regarding the proposed method's performance and model complexity. Lastly, the paper overlooks an analysis of critical parameters, such as the number of segments and the number of words per segment, and their impact on captioning quality. Addressing these points through minor revisions would strengthen the paper's contribution, and overall, it is recommended for acceptance.

In the paper titled "Show, Tell and Summarize: Dense Video Captioning Using Visual Cue Aided Sentence Summarization," authors Zhiwang Zhang et al. present a novel framework for dense video captioning, divided into two key stages: division and summarization[8]. The division stage employs existing image/video captioning methods to generate multiple sentence descriptions for each event proposal in untrimmed videos, while the summarization stage formulates dense video captioning as a visual cue aided sentence summarization problem. To achieve this, the paper proposes a two-stage LSTM network with a hierarchical attention mechanism, leveraging both visual and textual features to generate descriptive sentences for event proposals. The method is extensively evaluated on the ActivityNet Captions dataset, demonstrating superior performance over state-of-the-art approaches according to various evaluation metrics. The strengths lie in the innovative perspective of treating dense video captioning as a summarization challenge and the introduction of a hierarchical attention mechanism to capture temporal evolution within event proposals.

While the paper showcases several strengths, some notable weaknesses exist. Firstly, the absence of a comparison with other sentence summarization methods, such as LexRank, limits a comprehensive understanding of the proposed method's performance relative to existing summarization approaches. Additionally, the lack of qualitative analysis or examples of generated captions hinders the illustration of the method's advantages and limitations. Furthermore, the paper overlooks discussions on the computational complexity or efficiency of the proposed framework,

which is crucial for practical applications. Addressing these aspects through further comparative analysis, qualitative insights, and discussions on computational considerations would enhance the overall contribution of the paper.

Jiehang Xie Et al[9]. introduced a novel framework that exploits multimodal information and aesthetic guidelines for generating high-quality and engaging narrative video summaries. The paper is commendable for its well-crafted structure, clarity, and coherence. The proposed method, offering originality and innovation, stands out by addressing challenges inherent in existing video summarization approaches. The extensive experiments and user studies conducted to evaluate the proposed method's effectiveness and user satisfaction, comparing it against various baselines and manual summarization, underscore the robustness of the approach. Overall, the paper marks a significant advancement in video summarization and holds promise for inspiring future research directions and practical applications within the multimedia domain.

While the paper excels in several aspects, there is room for improvement. Providing additional details and in-depth analysis regarding the datasets used in the experiments, including sources, genres, lengths, and characteristics of videos, as well as the criteria and process for collecting and annotating subtitles and descriptions, would enhance the transparency and reproducibility of the study. Furthermore, addressing the limitations and challenges of the proposed method, such as scalability, robustness, and generalization across diverse video types and user preferences, along with discussions on the trade-offs between multimodal information and aesthetic guidelines, would contribute to a more comprehensive understanding of the approach. Finally, the inclusion of concrete examples showcasing generated video summaries alongside corresponding input videos, subtitles, and procedural texts would visually illustrate the results and effects of the proposed method, offering a clearer insight into its practical implications. Incorporating these suggestions would further strengthen the paper's impact and relevance in the domain of narrative video summarization.

Nayu Liu Et al[10]. introduced two innovative models for multimodal abstractive summarization of open-domain videos, employing different fusion strategies

and a fusion forget gate module to integrate video and audio transcript features. The evaluation on How2 and How2-300 h datasets reveals the models' superior performance, particularly in scenarios with noisy ASR transcripts. The provision of an ASR transcript dataset for How2 adds significant value to the research community. The paper is well-structured, clearly articulating the motivation, methodology, and results of the proposed models. The comprehensive experiments, including human evaluations and case studies, substantiate the efficacy and robustness of the proposed models, making a noteworthy contribution to multimodal abstractive summarization.

Providing more details on the implementation of the ASR system, including specifics on speech segmentation, ASR model architecture, and hyperparameters, would improve the reproducibility and understanding of the experiments. Additionally, comparing the proposed models with recent methods for video captioning or summarization that leverage multimodal features and attention mechanisms could provide a broader context for evaluating their relative strengths. To deepen insights into the proposed models, an analysis of the impact of different modalities and fusion methods on summary quality through ablation studies or qualitative analyses of examples would enrich the paper. Finally, discussing potential limitations and challenges, such as scalability to longer videos, generalization to diverse domains, and handling complex multimodal information, would further contribute to the paper's completeness. Addressing these suggestions would fortify the paper's contribution and broaden its impact in the field of multimodal abstractive summarization.

2.2 Audio To Text Summarization

The paper by Sushant Gautam et al. addresses the task of soccer game summarization using a multifaceted approach involving audio commentary, metadata, and captions[11]. Notably, the paper contributes by extending and curating soccer datasets with ground truth summaries, game metadata, and translations, fostering accessibility for the broader research community. The design and implementation of an end-to-end pipeline, incorporating diverse input modalities and natural language

processing tools, showcase the paper's commitment to a comprehensive summarization approach. The presentation of preliminary results from a comparative analysis of summarization methods within the pipeline, coupled with insightful discussions on challenges and future directions, further establishes the paper's contribution. Its well-structured organization, clarity, and emphasis on the novelty of the task, as well as the utility of datasets and software tools, underscore the paper's strengths in advancing soccer game summarization.

Despite its merits, the paper could benefit from additional details in the evaluation methodology, particularly concerning the number and characteristics of human evaluators, criteria for subjective ratings, and the statistical significance tests performed on the results. Incorporating qualitative examples of generated summaries, along with comparisons to ground truth and state-of-the-art methods, would enhance the paper's illustration of the proposed pipeline's strengths and weaknesses. Furthermore, an exploration of ethical and social implications, such as the impact on human journalism and potential biases arising from diverse data sources and models, would enrich the paper's discussion. Addressing these points would contribute to a more comprehensive understanding and further strengthen the significance of the presented soccer game summarization framework.

Murad Ali Khan and others[12]. presented a dual speech/text encoder model designed for summarizing lengthy spoken documents, extending up to 10 minutes. The proposed approach incorporates memory-efficient encoders like FNet and Informer to address the computational and memory challenges associated with processing extended speech sequences. The evaluation on the How2 dataset demonstrates the effectiveness of the model, showcasing improved performance over baseline models across various metrics. The paper is commendable for its clear and well-organized structure, providing a compelling motivation, precise problem formulation, and a comprehensive experimental setup. The dual encoder model marks a significant contribution to the field of speech summarization, particularly in handling the intricate task of summarizing lengthy spoken documents by leveraging both speech and text modalities.

While the paper stands out in several aspects, there are opportunities for im-

provement. To enhance the reproducibility and understanding of the proposed model, providing more details on data preprocessing, the hyperparameters used, and conducting ablation studies would be beneficial. Additionally, a comparative analysis with existing methods for speech summarization, such as pointer-generator networks or reinforcement learning approaches, would strengthen the paper's contribution by establishing its performance relative to established benchmarks in the field. Incorporating these elements would further solidify the paper's standing in the realm of speech summarization and contribute to its broader impact.

The paper titled "Leveraging large text corpora for end-to-end speech summarization" addresses the limitations of the traditional cascade approach in speech summarization by proposing two novel methods for end-to-end speech summarization (E2E SSum)[13]. The authors introduce a text-to-speech (TTS) based method and a TTS-free method, both leveraging a large external text summarization dataset for training. The motivation to tackle issues like ASR error propagation and the lack of nonverbal and acoustical information is well-founded, and the proposed methods demonstrate state-of-the-art performance on the How2 dataset, a significant contribution to the field of multimodal language understanding. The paper's comprehensive overview, clarity in presenting the methods and results, and thorough analysis make it a valuable contribution to the advancement of E2E SSum techniques.

A more in-depth exploration of the nuances, intricacies, and potential trade-offs involved in these methods would enhance the reader's understanding. Additionally, the paper acknowledges the limitations, such as domain mismatches in external corpora and a lack of diversity in speakers for synthesized speech, but could delve further into the potential impact of these factors on the generalization of the proposed methods. Further exploration of the proposed models' scalability and adaptability to diverse datasets or domains would contribute to a more comprehensive understanding of their applicability. Addressing these aspects would further solidify the paper's standing in the domain of E2E speech summarization and provide valuable insights for future research directions.

2.3 Text To Text Summarization

The paper introduces a compelling framework for abstractive summarization titled ”Fact-Driven Abstractive Summarization by Utilizing Multi-Granular Multi-Relational Knowledge[14].” The innovative approach leverages multi-granular factual information extracted from the source text, utilizing a fact-driven graph attention network and a hybrid pointer network for effective information retrieval and incorporation into the generated summaries. A notable strength lies in the novel modeling of fine-grained factual information within a graph structure, promising heightened informativeness and faithfulness in the generated summaries. Additionally, the paper introduces a novel fact-checking evaluator, enhancing the credibility of the proposed framework by verifying the factual consistency of phrase-level facts and facets. The empirical evaluation demonstrates the superiority of the proposed approach over state-of-the-art methods, as evidenced by improved ROUGE scores and enhanced factual correctness on two benchmark datasets.

Despite its strengths, the paper exhibits certain limitations. The lack of detailed insights into the construction of the factual graph and the integration of the graph attention network with the BART model impedes a comprehensive understanding of the proposed framework’s inner workings. Furthermore, the absence of comparisons with other graph-based or fact-based summarization methods hinders benchmarking and assessing the framework’s relative performance. Addressing these gaps through more detailed technical explanations, illustrative examples, and comparative analyses would enhance the paper’s contribution. Additionally, the paper could benefit from an exploration of the impact of different types of facts and facets on summarization quality, fostering a deeper understanding of the nuances in leveraging multi-granular multi-relational knowledge for abstractive summarization.

The paper, titled ”Multi-document summarization using selective attention span and reinforcement learning,” introduces the REISA model, a novel approach for abstractive multi-document summarization[15]. By leveraging reinforced selective attention span and dual reward functions, REISA aims to generate more co-

herent, faithful, and abstractive summaries compared to existing baselines. The recalibration of attention weights and optimization of both syntactic and semantic rewards contribute to the model's effectiveness. The evaluation on benchmark datasets, Multinews and CQASumm, showcases significant improvements in key metrics such as ROUGE, BERTScore, and QAEval. The paper's clarity, organization, and comprehensive experimental results, including human evaluations and ablation studies, strengthen its contributions to the field of multi-document summarization. The discussion on limitations and future directions further enhances the paper's overall impact.

While the paper demonstrates notable strengths, there are areas for improvement. Providing more detailed insights into the implementation and training of the reinforced attention span, with a clear distinction from other attention mechanisms, would enhance the technical understanding of the proposed model. Additionally, expanding the comparison to include more recent and relevant baselines, such as LONGT5 and Topic-Guided MDS, could provide a more nuanced understanding of REISA's performance in the current landscape of transformer-based models for multi-document summarization. Performing more in-depth qualitative analysis, including examining failure cases and providing additional examples of attention recalibration and reward functions, would offer a richer understanding of the model's behavior and potential limitations. Addressing these suggestions would contribute to a more thorough evaluation and interpretation of the REISA model's capabilities and limitations in the context of multi-document summarization.

Guangsheng Bao Et al[16]. proposed a novel method for text summarization by proposing a contextualized rewriting framework. The authors leverage group-tags to align extractive sentences with summary sentences, allowing the rewriter to consider both document and summary context during the rewriting process. The general seq2seq model with group-tag alignments is demonstrated through three rewriter instances based on various pre-trained models. The paper's strength lies in addressing the challenging task of enhancing extractive summaries by incorporating contextual information, and the introduction of group-tags provides a novel means to represent the alignment between extractive and summary sentences. The general

framework's flexibility across different seq2seq models and extractors showcases its adaptability, and the comprehensive evaluation on the CNN/DailyMail dataset, with improvements in ROUGE scores and human ratings, demonstrates the effectiveness of the proposed method.

The generation process of group-tags for both input documents and output summaries lacks detailed explanation, particularly regarding sentences not selected by the extractor or those subjected to splitting or merging during rewriting. Providing examples and outlining the rules or algorithms for generating group-tags would enhance clarity. Additionally, the paper would benefit from a thorough discussion of the method's limitations and challenges. Considerations such as robustness to noisy or inaccurate extractive summaries, handling complex or lengthy documents, and addressing the trade-off between informativeness and conciseness during rewriting should be explored. Furthermore, a comparative analysis with other related works that utilize contextual information for text summarization would provide insights into the distinct contributions and potential synergies of the proposed method. Addressing these aspects would contribute to a more comprehensive understanding of the proposed contextualized rewriting framework and its implications for text summarization.

CHAPTER 3

PROBLEM DEFINITION

The escalating number of content creators contributes to the overwhelming volume of videos being produced, making it increasingly challenging for users to find the information they seek. Creators, in their pursuit of audience engagement, may resort to attention-grabbing techniques such as click-bait titles and thumbnails, further complicating the search for relevant content.

3.1 Problem Formulation

The challenge at hand involves the development of an innovative application capable of autonomously summarizing the transcripts of YouTube videos, with a paramount emphasis on preserving the original meaning of the text. In the contemporary landscape of digital content, where a vast array of videos are continuously generated and shared online, the need for efficient and accurate information retrieval is crucial. The objective is to create a tool that can discern and distill the essential information from video transcripts, presenting users with concise summaries that encapsulate the key elements without sacrificing the nuanced meaning of the original content. The application seeks to address the common frustration experienced by users in navigating through lengthy videos to find specific information, exacerbated by misleading elements such as thumbnails, descriptions, and promotional content.

3.2 Problem Statement

The following describes the work's problem statement:

"Develop an application that can automatically summarize the transcripts of YouTube videos, without distorting the actual meaning of the text."

3.3 Objectives

The outlined objectives aim to address various challenges associated with accessing and extracting meaningful information from YouTube videos. Let's delve deeper into each objective:

1. To implement an algorithm to generate summaries of the transcripts.
2. To provide the user with a summary of the transcript that contains the key information and preserves the meaning of the original text.
3. To reduce the time and effort of the user in finding information from videos.

CHAPTER 4

PROPOSED METHODOLOGY

4.1 Overall Data Flow

The proposed method outlines a systematic approach to enhance the user experience in accessing information from YouTube videos using a web application as in Fig 4.1. The primary objectives include retrieving video transcripts, implementing an algorithm for summarization, providing users with meaningful summaries, and reducing time and effort in information retrieval.

In the initial phase, users input a YouTube video link as the starting point for information extraction. The application then utilizes the YouTubeTranscriptApi to retrieve the transcript of the specified video. This ensures that the raw textual content of the video is accessible for further processing. The next step involves the implementation of an algorithm for summarizing the transcripts. The chosen method employs Gemini-pro to generate detailed notes from the transcript. The detailed note passes to the fine-tuned Pegasus model for retrieving the shortest summary. This algorithm is crucial in generating concise and meaningful summaries while preserving the original meaning of the content.

Following the summarization process, the application aims to reduce user effort and time in finding relevant information within videos. By presenting users with summarized transcripts, the application streamlines the content consumption process, allowing users to quickly grasp key details without the need to watch the entire video.

In terms of user interaction, the application offers flexibility by providing users with options to either view the summarized text or listen to an audio version of the summary. This accommodates different user preferences and ensures accessibility for a diverse audience.

In summary, the proposed method combines technological tools and algorithms to create an application that simplifies the extraction of valuable information from YouTube videos. By integrating transcript retrieval, advanced summarization techniques, and tokenization, the application strives to optimize the user experience, making content consumption more efficient and user-friendly.

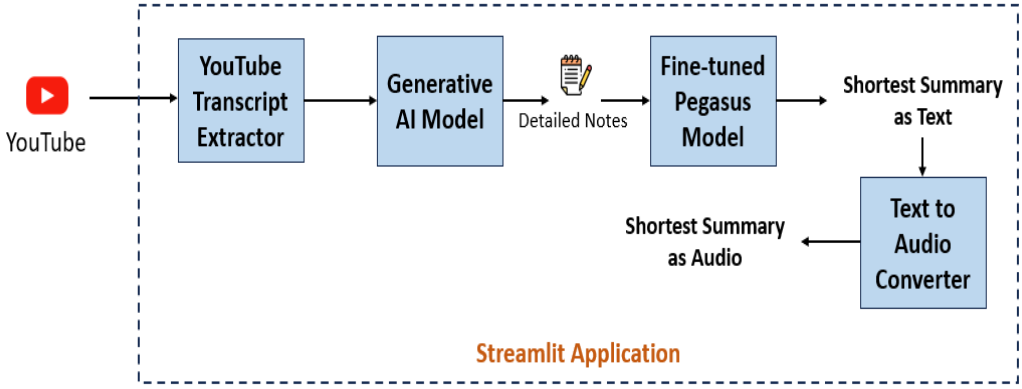


Figure 4.1: Design of the application

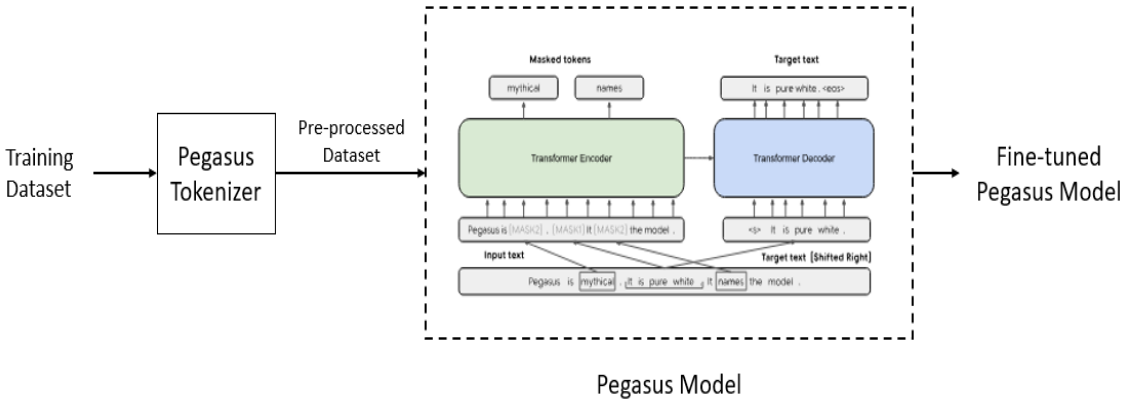


Figure 4.2: Architecture of Fine-tuned Pegasus

4.2 Data Collection

The new dataset was created by combining the **SAMSum dataset** and the **DialogSum Corpus**. The SAMSum dataset comprises approximately 16,000 messenger-like conversations, each accompanied by a summary. These conversations were crafted by linguists fluent in English, who were instructed to replicate the kinds of conversations they typically engage in, mirroring the proportion of topics found in their actual messaging habits. The conversations vary in style and register, ranging from informal to formal, and may include slang, emoticons, and typos. Summaries were then added, providing a concise third-person account of each conversation.

The DialogSum Corpus is a comprehensive resource for dialogue summarization research. It includes dialogues that capture a wide range of everyday conversational scenarios. These interactions cover diverse topics such as education, work, healthcare, shopping, leisure, and travel. The dataset features conversations set in various real-life contexts, involving friends, colleagues, customers, and service providers.

The new dataset is divided into training, testing, and validation files, each containing two features: ”dialogues” and ”summary.” These CSV files are then converted into a dictionary format to facilitate preprocessing.

```
DatasetDict({
    train: Dataset({
        features: ['dialogue', 'summary'],
        num_rows: 27192
    })
    test: Dataset({
        features: ['dialogue', 'summary'],
        num_rows: 2319
    })
    validation: Dataset({
        features: ['dialogue', 'summary'],
        num_rows: 1318
    })
})
```

Figure 4.3: Dataset architecure

```
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.
```

Figure 4.4: Sample Data

4.2.1 Preprocessing

The preprocessing of the dataset ensures that the inputs are in string format, tokenizes the dialogues and summaries with specified maximum lengths and truncation, and returns the tokenized inputs, attention masks, and target labels. The function is then applied to the entire dataset in batches, resulting in a tokenized dataset ready for model training or evaluation.

4.3 Design of The Web Application

In this proposed web application, users are presented with an intuitive interface that facilitates the input of YouTube video links, streamlining the process of extracting valuable information from selected videos. The underlying YouTube Video API seamlessly retrieves transcripts, ensuring a swift and efficient operation. The application prioritizes user convenience by supporting various video link formats and implementing robust validation mechanisms. Following transcript extraction, a sophisticated text summarization module comes into play. This module employs an effective algorithm that condenses lengthy transcripts while preserving essential information. This feature enhances the user experience by providing concise

and relevant summaries. To cater to diverse user preferences, the web application utilizes Google Text to Speech (GTTS) for converting text summaries into audio.

4.3.1 Streamlit

Streamlit is an open-source Python framework designed specifically for data scientists and AI/ML engineers to create interactive, web-based data applications quickly and efficiently. It allows users to build and share custom applications with minimal effort, using only a few lines of code. The following are the key features of the streamlit:

1. Simplicity and Ease of Use:
    - Minimal Code: Streamlit enables the creation of data apps using simple Python scripts. There is no need for front-end programming languages like HTML, CSS, or JavaScript.
    - Auto-Reload: When you save your code, Streamlit automatically reloads your app, so you can see changes in real-time.
2. Interactive Widgets:
    - Widgets Integration: Streamlit provides a variety of widgets such as sliders, buttons, and text inputs that allow users to interact with the data. These widgets are easy to implement and integrate directly with your Python code.
    - Dynamic Updates: Widgets can trigger updates in real-time, enabling dynamic interactions within the app.
3. Data Visualization:
    - Built-in Support: Streamlit supports many popular data visualization libraries like Matplotlib, Plotly, Altair, and Bokeh. You can directly embed these visualizations into your Streamlit app.

- Customization: Visualizations can be customized and updated based on user inputs and interactions.

4. Seamless Data Handling:

   - File Uploads: Users can upload files (e.g., CSVs, images) directly through the app interface, which can then be processed and analyzed within the app.
   - Data Manipulation: Streamlit makes it easy to read, manipulate, and display data in various formats, including data frames and tables.

4.3.2 YouTube Transcript API

The YouTube Transcript API serves as a pivotal component in our web application, facilitating the extraction of textual content from YouTube videos. Leveraging this API, we seamlessly retrieve the transcripts associated with specified video links. The integration ensures a streamlined and efficient process, allowing users to effortlessly access and work with the spoken content within the videos.

Through the YouTube Transcript API, our application taps into the extensive YouTube database, retrieving accurate and comprehensive transcripts. This functionality enhances the user experience, enabling the application to deliver precise textual representations of the audio content present in the selected videos.

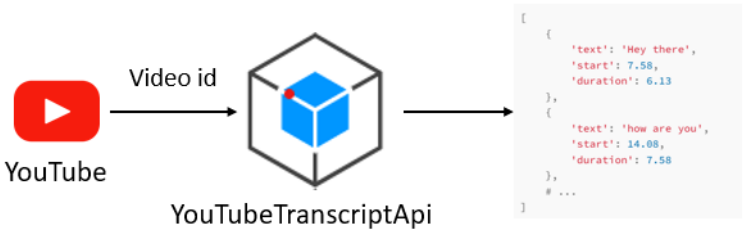


Figure 4.5: YouTube transcript extraction

4.3.3 Gemini-pro Generative AI model

Gemini Pro is a cutting-edge large language model (LLM) developed by Google DeepMind, part of the Gemini family of generative AI models. Designed to handle

both text and images as input, Gemini Pro excels in a wide array of tasks, making it a versatile tool for developers. Gemini Pro represents a significant advancement in the realm of AI, offering a comprehensive suite of features that cater to diverse applications. Whether it's natural language processing, content creation, or coding, Gemini Pro stands out as a powerful tool designed to meet the complex needs of developers and AI/ML engineers. Its robust performance, coupled with a commitment to safety and ethical use, makes Gemini Pro an exceptional choice for a wide range of AI-driven tasks. Here's an in-depth look at Gemini Pro, its features and capabilities:

1. Multimodal Input Handling:

    - Gemini Pro can process both text and image inputs, enabling a richer interaction and understanding of content. This multimodal capability allows it to integrate visual and textual information seamlessly.

2. Natural Language Processing:

    - **Summarization:** Gemini Pro can create concise summaries of lengthy documents, retaining essential information. This is useful for condensing chapters of textbooks or generating product descriptions from detailed texts.
    - **Question Answering:** The model can provide accurate answers to questions based on the text, automating the creation of FAQs and knowledge base articles.
    - **Classification:** Gemini Pro can assign labels to text, such as indicating grammatical correctness or categorizing content by topic or sentiment.
    - **Sentiment Analysis:** This form of classification identifies the emotional tone of text, labeling it as positive, or negative, or specifying sentiments like anger or happiness.
    - **Entity Extraction:** The model can identify and extract specific information from text, such as names of movies or people mentioned in an article.

- **Content Creation:** Gemini Pro generates text based on specified requirements and contexts, such as drafting emails in a particular tone or style.

3. Advanced Coding Capabilities:

   - **Code Generation:** Gemini Pro excels in generating high-quality code across various programming languages, making it an invaluable tool for developers. It can also assist in explaining and debugging code.

4. Sophisticated Reasoning:

   - The model demonstrates advanced reasoning abilities, making it adept at extracting insights from large datasets and contributing to breakthroughs in various research fields.

5. Reliable and Efficient Performance:

   - Trained using Google's Tensor Processing Units (TPUs), Gemini Pro is highly reliable, scalable, and efficient. It operates significantly faster than its predecessors, ensuring smooth performance even with complex tasks.

6. Safety and Responsibility:

   - Comprehensive safety evaluations are a cornerstone of Gemini Pro's development. Google DeepMind addresses potential risks, incorporating safeguards against biases and toxicity. The development team collaborates with external experts to ensure ethical use and adherence to responsible AI practices.

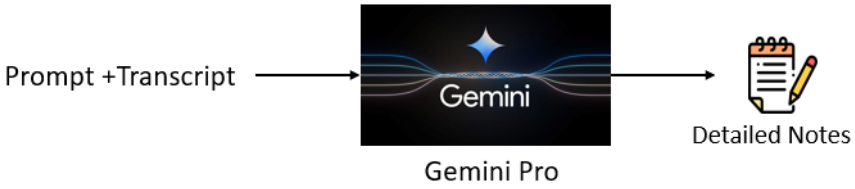


Figure 4.6: Detailed note generation using Gemini-pro

4.3.4 Pegasus Model

Pegasus is a state-of-the-art (SOTA) model for text summarization, leveraging a pretraining task specifically designed to mirror the summarization process. In Pegasus, key sentences are intentionally removed or masked from an input document and are subsequently generated together as a single output sequence from the remaining sentences, akin to an extractive summary[17]. This unique pretraining task enables Pegasus to achieve exceptional summarization performance across all 12 evaluated downstream tasks, as validated by both ROUGE metrics and human evaluations. The model, contributed by sshleifer, demonstrates its robustness and efficacy in the field of natural language processing.

Pegasus operates as a sequence-to-sequence model with an encoder-decoder architecture, similar to the BART model. It is pre-trained on two self-supervised objective functions: Masked Language Modeling (MLM) and a novel summarization-specific pretraining objective called Gap Sentence Generation (GSG). In MLM, encoder input tokens are randomly replaced by mask tokens, which the encoder must then predict, a mechanism akin to BERT. GSG, on the other hand, involves replacing entire encoder input sentences with a second mask token, which is then fed to the decoder. The decoder employs a causal mask to conceal future words, functioning like a traditional autoregressive transformer decoder.

In terms of implementation, Pegasus models are transformer encoder-decoders featuring 16 layers in each component. The implementation closely follows that of BartForConditionalGeneration, with a few key configuration differences. Pegasus uses static, sinusoidal position embeddings and begins generation with the pad_token_id, which has a token embedding of zero, serving as the prefix. Additionally, Pegasus employs a greater number of beams (num_beams=8) during the generation process to enhance output quality.

4.3.4.1 Fine-tuning of the Pegasus Model

In this project, the pegasus model is fine-tuned using Samsum and Dialogsum Dataset. We specified the model checkpoint for the PEGASUS model. It then loaded the corresponding tokenizer for this model checkpoint. Subsequently, it

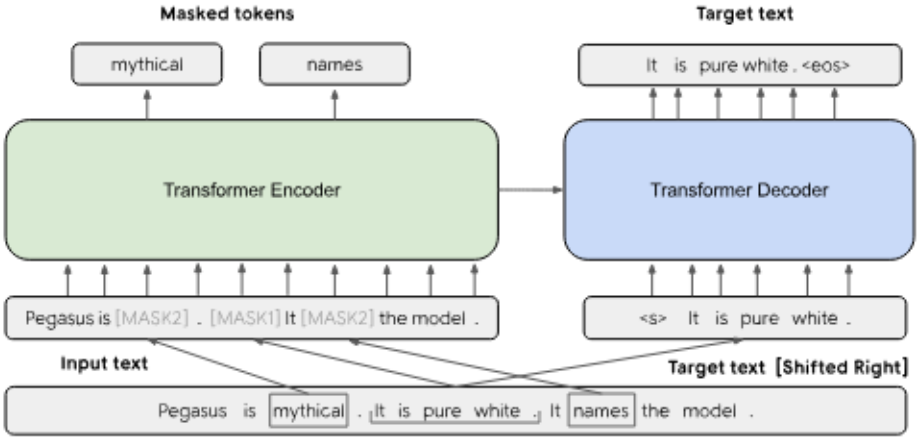


Figure 4.7: The architecture of Pegasus model

loaded the PEGASUS model itself and moved it to the appropriate computation device, either GPU or CPU, based on availability. The following figure shows that the code sets up the training environment for the PEGASUS model fine-tuned on the CNN/DailyMail dataset using the transformers library. It first imports and initializes necessary components, including a data collator for padding sequences and 'tokenizer' for text processing. The training configuration is specified through 'TrainingArguments', detailing parameters such as batch size, number of epochs, evaluation strategy, and gradient accumulation steps. Finally, the Trainer is initialized with the model, tokenizer, data collator, and training and evaluation datasets, readying the setup for model training and evaluation.

```
from transformers import TrainingArguments, Trainer

trainer_args = TrainingArguments(
    output_dir='pegasus-ami-corpus-samsum', num_train_epochs=8, warmup_steps=500,
    per_device_train_batch_size=1, per_device_eval_batch_size=1,
    weight_decay=0.01, logging_steps=10,
    evaluation_strategy='steps', eval_steps=500, save_steps=1e6,
    gradient_accumulation_steps=16
)
```

```
/opt/conda/lib/python3.10/site-packages/transformers/training_args.py:1474: FutureWarning: `evaluation_strategy` is d
eprecated and will be removed in version 4.46 of 🤗 Transformers. Use `eval_strategy` instead
  warnings.warn(
Using the `WANDB_DISABLED` environment variable is deprecated and will be removed in v5. Use the --report_to flag to
control the integrations used for logging result (for instance --report_to none).
```

+ Code + Markdown

```
trainer = Trainer(model=model_pegasus, args=trainer_args,
                  tokenizer=tokenizer, data_collator=seq2seq_data_collator,
                  train_dataset=dataset_ami_corpus_samsum_pt["train"],
                  eval_dataset=dataset_ami_corpus_samsum_pt["validation"])
```

Figure 4.8: Code for training Pegasus

4.3.5 Text to Speech Conversion

Google Text-to-Speech (GTTS) is a service provided by Google that converts text input into spoken audio. In the context of your web application, GTTS is employed to transform the summarized text generated from YouTube video transcripts into an audio format. It seamlessly transforms the summarized text into natural-sounding audio through sophisticated speech synthesis algorithms. This conversion process not only ensures a human-like quality but also offers a personalized experience with customization options for language, voice, and speech speed. The resulting audio file can be played back instantly or saved for future use, catering to users who prefer auditory content over traditional reading. GTTS significantly contributes to the accessibility of the application, providing an alternative means for users with visual impairments or those who favor auditory consumption.

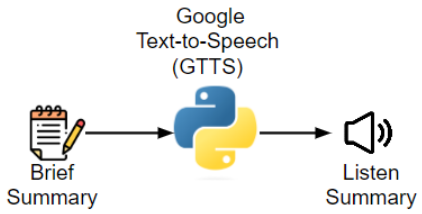


Figure 4.9: Google Text-to-Speech (GTTS)

CHAPTER 5

EVALUATION RESULTS AND DISCUSSIONS

This project focused on fine-tuning the Pegasus model on the Kaggle platform. The primary objective was to enhance the model's summarization capabilities using a combined dataset from the SAMSum and DialogSum Corpus. The dataset used for this project consisted of 27,192 entries, each featuring the following components: dialogue, summary, input_ids, attention_mask, and labels. These features were essential for preparing the data, ensuring that the model could effectively learn from and summarize the given dialogues. Tried different numbers of epochs for training. Table 5.1 shows the results of different epochs. The epoch which has the lowest training loss is chosen as the final.

Epochs	Training Loss
1	1.5733
3	1.4343
5	1.3610
8	1.1627
10	NA

Table 5.1: Epochs and Training losses

The fine-tuning process was carried out over 8 epochs, taking approximately 10 hours to complete. During this training period, the model's parameters were adjusted to minimize the loss function, leading to a final training loss of 1.16274. The extended training duration and iterative learning process helped in refining the

model's performance. To assess the effectiveness of the fine-tuned Pegasus model, it was tested using a single data entry from the 'test' set. The outcome of this test is depicted in the following figure, which illustrates the model's capability to generate detailed and accurate summaries from the input dialogues.

```
TrainOutput(global_step=13592, training_loss=1.162744076200062, metrics={'train_runtime': 39278.8779, 'tra
in_samples_per_second': 5.538, 'train_steps_per_second': 0.346, 'total_flos': 1.0226116206165197e+17, 'tra
in_loss': 1.162744076200062, 'epoch': 7.997646366578405})
```

Figure 5.1: Training Output

```
Dialogue:
Hannah: Hey, do you have Betty's number?
Amanda: Lemme check
Hannah: <file_gif>
Amanda: Sorry, can't find it.
Amanda: Ask Larry
Amanda: He called her last time we were at the park together
Hannah: I don't know him well
Hannah: <file_gif>
Amanda: Don't be shy, he's very nice
Hannah: If you say so..
Hannah: I'd rather you texted him
Amanda: Just text him ????
Hannah: Urgh.. Alright
Hannah: Bye
Amanda: Bye bye

Reference Summary:
Hannah needs Betty's number but Amanda doesn't have it. She needs to contact Larry.

Model Summary:
Amanda can't find Betty's number. Larry called her the last time they were at the park together. Hannah wa
nts Amanda to text Larry instead.
```

Figure 5.2: Testing Output

5.1 ROUGE Score to Evaluate Summaries

ROUGE, which stands for Recall-Oriented Understudy for Gisting Evaluation, is a set of metrics used to evaluate the performance of automatic text summarization and machine translation systems. ROUGE compares an automatically generated summary or translation against a set of reference summaries, typically created by humans, to measure its quality.

In essence, recall in the context of ROUGE measures how well the system-generated summary captures the content of the reference summary. For example, if we focus on individual words, recall can be computed as the proportion of words in the reference summary that also appear in the system-generated summary.

ROUGE-N, ROUGE-S, and ROUGE-L can be thought of as the granularity of texts being compared between the system summaries and reference summaries[18].

- **ROUGE-N** — measures unigram, bigram, trigram and higher order n-gram overlap.

$$ROUGE - N = \frac{\sum_{S\epsilon\{ReferenceSummaries\}} \sum_{gram_n \epsilon S} Count_{match}(gram_n)}{\sum_{S\epsilon\{ReferenceSummaries\}} \sum_{gram_n \epsilon S} Count(gram_n)} \quad (1)$$

- **ROUGE-L** — measures longest matching sequence of words using LCS. An advantage of using LCS is that it does not require consecutive matches but in-sequence matches that reflect sentence level word order. Since it automatically includes longest in-sequence common n-grams, you don't need a predefined n-gram length.

- **ROUGE-S** — Is any pair of words in a sentence in order, allowing for arbitrary gaps. This can also be called skip-gram concurrence. For example, skip-bigram measures the overlap of word pairs that can have a maximum of two gaps in between words. As an example, for the phrase "cat in the hat" the skip-bigrams would be "cat in, cat the, cat hat, in the, in hat, the hat".

The model's ROUGE score is calculated using the test set of the dataset. In order to calculate ROUGE scores we need to understand Recall and Precision in text. Following are the equations to calculate Precision, Recall, and F1 scores (Harmonic mean):

$$PRECISION = \frac{Overlapping\ \ number\ \ of\ \ n{-}grams}{Number\ \ of\ \ n{-}grams\ \ in\ \ the\ \ candidate} \quad (2)$$

$$RECALL = \frac{Overlapping\ \ number\ \ of\ \ n{-}grams}{Number\ \ of\ \ n{-}grams\ \ \ \ \ \ \ \ in\ \ the\ \ reference} \quad (3)$$

$$F1 = \frac{2{\times}Precision{\times}Recall}{Precision{+}Recall} \quad (4)$$

Following are the Rouge scores obtained for the test dataset after training the model for 8 epochs:

ROUGE Scores	Precision	Recall	F1 Score
Rouge-1	0.7692	0.7143	0.7407
Rouge-2	0.6000	0.5556	0.5769
Rouge-L	0.7692	0.7143	0.7407
Rouge-Lsum	0.7692	0.7143	0.7407

Table 5.2: ROUGE Scores

5.1.1 Frontend Development using Streamlit

In this project, Streamlit was used to develop the UI of the application. Streamlit, an open-source Python framework, enables data scientists and AI/ML engineers to quickly build and deploy dynamic data apps with minimal code. Users can input a YouTube video link into a text box, and upon clicking enter, the application's interface will display the video's thumbnail. By clicking the 'get details' button, the application generates and shows both a detailed summary and the shortest summary of the video. In addition, users have the option to listen to the shortest summary. The summary generation process is efficient, producing results within seconds.

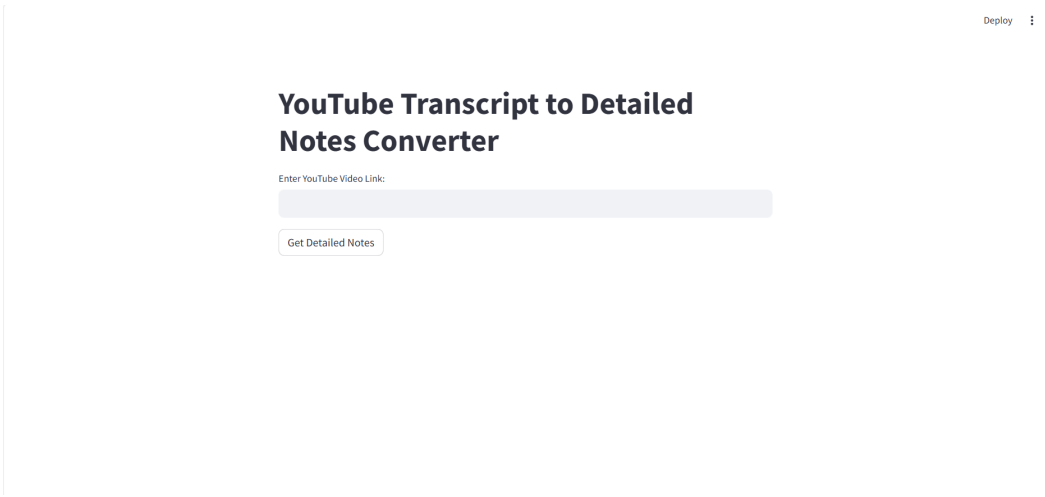


Figure 5.3: Web Aplication UI

Figure 5.4: Web application screenshot 1

Figure 5.5: Web application screenshot 2

CHAPTER 6

CONCLUSION

In conclusion, the proposed method presents a comprehensive and effective solution to address the challenges associated with accessing information from YouTube videos. By systematically integrating key steps, including transcript retrieval, advanced summarization algorithms, and tokenization, the application aims to significantly enhance the user experience. The utilization of the YouTubeTranscriptApi ensures seamless access to video transcripts, laying the foundation for subsequent processing. The successful fine-tuning of the Pegasus model on Kaggle, using the combined SAMSum and DialogSum Corpus dataset, underscores the model's enhanced summarization abilities. The rigorous training process, along with the comprehensive evaluation, demonstrates that the model is well-prepared for deployment and further testing. The results indicate a promising direction for future applications of the Pegasus model in automated summarization tasks.

The proposed method places a significant emphasis on enhancing user efficiency and minimizing time investment. By providing summarized transcripts, users can swiftly access crucial information without navigating through lengthy videos, addressing challenges like misleading content and extended promotional segments. This aligns with the contemporary need for efficient information consumption. Offering users the option to view summarized text or listen to an audio version reflects a user-centric approach, enhancing overall accessibility and usability.

PUBLICATIONS

1. Evelin Manoj, Rahmathulla K **"YOUTUBE VIDEO TRANSCRIPT SUMMARIZATION USING DEEP LEARNING",** International Conference on Recent Advancements in Engineering and Technology (ICORETech), LBS COLLEGE OF ENGINEERING, KASARAGOD, 2024. (Under Review)

REFERENCES

[1] S. Sharma, G. Aggarwal, and B. K. Rai, "A survey on the dataset, techniques, and evaluation metric used for abstractive text summarization," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 22, no. 3, pp. 681–689, 2024.

[2] P. Verma and A. Verma, "A review on text summarization techniques," *Journal of scientific research*, vol. 64, no. 1, pp. 251–257, 2020.

[3] N. Moratanch and S. Chitrakala, "A survey on extractive text summarization," in *2017 International Conference on Computer, Communication and Signal Processing (ICCCSP)*, pp. 1–6, 2017.

[4] V. Mane, K. Dupare, P. Dupare, T. Ganvir, and R. Ghode, "Youtube transcript summarizer using huggingface transformer,"

[5] S. Pesaru, B. Shravya, E. Devendhar, and A. Shashank, "Language alchemist–a youtube transcript summarizer,"

[6] A. Bhagat and P. Anjankar, "Text summarization on youtube videos in educational domain," 2023.

[7] Z. Zhang, D. Xu, W. Ouyang, and L. Zhou, "Dense video captioning using graph-based sentence summarization," *IEEE Transactions on Multimedia*, vol. 23, pp. 1799–1810, 2021.

[8] Z. Zhang, D. Xu, W. Ouyang, and C. Tan, "Show, tell and summarize: Dense video captioning using visual cue aided sentence summarization," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 9, pp. 3130–3139, 2020.

[9] J. Xie, X. Chen, T. Zhang, Y. Zhang, S.-P. Lu, P. Cesar, and Y. Yang, "Multimodal-based and aesthetic-guided narrative video summarization," *IEEE Transactions on Multimedia*, vol. 25, pp. 4894–4908, 2023.

[10] N. Liu, X. Sun, H. Yu, F. Yao, G. Xu, and K. Fu, "Abstractive summarization for video: A revisit in multistage fusion network with forget gate," *IEEE Transactions on Multimedia*, vol. 25, pp. 3296–3310, 2023.

[11] S. Gautam, C. Midoglu, S. Shafiee Sabet, D. B. Kshatri, and P. Halvorsen, "Soccer game summarization using audio commentary, metadata, and captions," in *Proceedings of the 1st Workshop on User-Centric Narrative Summarization of Long Videos*, NarSUM '22, (New York, NY, USA), p. 13–22, Association for Computing Machinery, 2022.

[12] T. Kano, A. Ogawa, M. Delcroix, R. Sharma, K. Matsuura, and S. Watanabe, "Speech summarization of long spoken document: Improving memory efficiency of speech/text encoders," in *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5, 2023.

[13] K. Matsuura, T. Ashihara, T. Moriya, T. Tanaka, A. Ogawa, M. Delcroix, and R. Masumura, "Leveraging large text corpora for end-to-end speech summarization," in *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5, 2023.

[14] Q. Mao, J. Li, H. Peng, S. He, L. Wang, P. S. Yu, and Z. Wang, "Fact-driven abstractive summarization by utilizing multi-granular multi-relational knowledge," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 1665–1678, 2022.

[15] Y. K. Atri, V. Goyal, and T. Chakraborty, "Multi-document summarization using selective attention span and reinforcement learning," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 31, pp. 3457–3467, 2023.

[16] G. Bao and Y. Zhang, "A general contextualized rewriting framework for text summarization," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 31, pp. 1624–1635, 2023.

[17] J. Zhang, Y. Zhao, M. Saleh, and P. J. Liu, "Pegasus: Pre-training with extracted gap-sentences for abstractive summarization," 2020.

[18] C.-Y. Lin, "ROUGE: A package for automatic evaluation of summaries," in *Text Summarization Branches Out*, (Barcelona, Spain), pp. 74–81, Association for Computational Linguistics, July 2004.