Multi-model Video Summarization

Periseti Sai Ram Mohan Rao,
Department of Computer Science Engineering
Indian Institute of Information Technology, Raichur,
Raichur, 584135, Karnataka, India
{cs20b1013@}@iiitr.ac.in

Neha Agarwal
Department of Computer Science Engineering
Indian Institute of Information Technology Raichur,
Raichur 584135, Karnataka, India
neha@iiitr.ac.in

***Abstract*— This project proposes a novel multi-model video summarization system that leverages the strengths of deep learning to create concise and informative summarizes videos. The system first employs BLIP, a bidirectional encoder that jointly learns image representations and language descriptions, to generate captions for each frame of a video. This comprehensive set of captions cap- tures the rich visual content of the video, providing a detailed description of the actions, objects, and scenes presented throughout. The ever-growing volume of video data uploaded and consumed online presents challenges in efficiently extracting key information. Manual video summarization is a time-consuming and laborious task, prompting the need for automated techniques. This research investigates multi-model video summarization, an approach that leverages the strengths of different modalities (audio and video) to create concise and informative summaries. Next, the system employs CLIP, a contrastive language-image pre-training model, to evaluate the relevance of each generated caption to the overall video content. CLIP's ability to assess the relationship between text and image representations allows the system to identify the most informative captions that best represent the key aspects of the video. By focusing on these high-scoring captions, the system ensures that the resulting summary is accurate and conveys the most important parts of the video content. Finally, the system utilizes a third deep learning model, a T5, BART, and PEGASUS Text Summarization Model, to generate a concise summary of the video. These text summarization models are specifically trained on large amounts of text data to condense information while preserving the core meaning effectively. By feeding the most relevant captions identified by CLIP into the summarization model, the system ensures that the generated summary accurately reflects the key ideas and events depicted in the video. This approach aims to capture the visual and textual aspects of the video, resulting in a more informative and comprehensive summary compared to traditional text-based summarization methods.**

***Index Terms*— BLIP, CLIP, BART, T5, PEGASUS, BLEU, ROUGE, BERTScore**

I. INTRODUCTION

The ever-increasing deluge of online video content creates a significant challenge: efficiently extracting the crucial information and insights it holds. Traditional automatic video summarization approaches, which primarily rely on text analysis, often fail to capture the richness of the content. Visual details, gestures, and scene changes can convey significant meaning that's absent from the spoken word. This project proposes a novel solution: Multi-Modal Video Summarization.

This system harnesses the power of both natural language processing (NLP) and computer vision (CV) techniques to create a more comprehensive understanding of video content. It begins by transcribing the audio track, meticulously capturing the spoken content. Next, it extracts keyframes at strategic intervals, providing a visual snapshot of the content at different points in the video. By leveraging state-of-the-art NLP models like T5, BART, and PEGASUS, the system generates summaries based on the transcribed text. To bridge the gap between spoken language and visual elements, it additionally utilizes cutting-edge image captioning models. These models analyze the keyframes, extracting meaningful descriptions that encapsulate the visual details.

This multi-modal approach strives to provide a richer and more informative understanding of video content. By weaving together textual and visual information, the summaries aim to capture the essence of the video, encompassing both the spoken language and the visual storytelling that unfolds on screen. This allows viewers to not only grasp the key points but also gain a deeper appreciation for the narrative conveyed through both words and images.

II. AIMS AND OBJECTIVES IN MULTI-MODEL VIDEO SUMMARIZATION:

This project aims to develop a system for generating summaries of multimedia content, specifically videos. The system will leverage the strengths of multiple deep learning models, including a speech recognition model for audio transcription, an image captioning model for video frame analysis, and a text summarization model for

compressRecent years have seen rapid advancements in computer vision and natural language processing. Still, many real-world problems are inherently multimodal - they involve several distinct forms of data, such as images and text. Visual-language models face the challenge of combining modalities so that they can open the door to a wide range of applications. Some of the image-to-text tasks that visual language models can tackle include image captioning, image-text retrieval, and visual question answering. Image captioning can aid the visually impaired, create useful product descriptions, identify inappropriate content beyond text, and more. Image-text retrieval can be applied in multimodal search, as well as in applications such as autonomous driving. Visual question-answering can aid in education, enable multimodal chatbots, and assist in various domain-specific information retrieval applications.

Modern computer vision and natural language models have become more capable; however, they have also significantly grown in size compared to their predecessors. While pre-training a single-modality model is resource-consuming and expensive, the cost of end-to-end vision-and-language pre-training has become increasingly prohibitive. BLIP-2 tackles this challenge by introducing a new visual-language pre-training paradigm that can potentially leverage any combination of pre-trained vision encoder and LLM without having to pre-train the whole architecture end to end. This enables achieving state-of-the-art results on multiple visual-language tasks while significantly reducing the number of trainable parameters and pre-training costs. Moreover, this approach paves the way for a multimodal ChatGPT-like model. ing the extracted text into a concise and informative summary. By combining these components, the system will be able to capture the salient information from both the audio and visual components of a video, resulting in summaries that are more comprehensive and faithful to the original content.

Automatic Speech Recognition (ASR): Transcribes the audio content of the video into text format. In this section, we describe the first step in processing the video. An ASR model is used to convert the spoken words in the video into written text. This creates a textual representation of the audio content.

Image Captioning using BLIP-2: Extracts captions from keyframes using a pre-trained image captioning model. In this section, we focus on the visual aspect of the video. A pre-trained image captioning model is employed to analyze keyframes (representative frames) extracted from the video. The model generates captions describing the content of each keyframe. These captions provide a textual understanding of the visual information in the video.

Evaluation of Captions using CLIP: Utilizing the insights gleaned from the multi-model inference pipeline, an evaluation employing the CLIP model offers a straightforward yet powerful means to assess the quality and relevance of generated captions. By harnessing CLIP's adeptness in understanding the semantic context of images and text, the evaluation process provides a nuanced understanding of the efficacy of the captioning system. Through a series of ranked captions juxtaposed with their corresponding images, CLIP unveils the intricate interplay between visual content and textual descriptions. This evaluation methodology serves as a robust benchmark, enabling stakeholders to gauge the accuracy and coherence of the generated captions in relation to the underlying images.

Fusion of Audio and Video Data: Integrates the audio and visual summaries into a single summary. This section combines the information obtained from the ASR (text from audio) and image captioning (text from keyframes). The process likely involves merging the two lists of strings (audio summary and video summary) into a single list. However, a simple string combination might not be sufficient. More sophisticated techniques like identifying coreference (referring to the same thing with different words) or topic modeling might be used to create a cohesive and informative summary.

Text Summarization: Summarizes the multi-fusion text of audio and image captions. This section takes the combined textual content from the previous step (audio and visual summaries) and generates a concise summary. A transformer-based summarization model (e.g., T5, BART, PEGASUS) is used for this task. These models are advanced deep learning architectures trained specifically to extract the most important information from a given text and present it in a shorter, more focused form. The chosen model will analyze the combined text from audio and visuals, identify key points, and generate a summary that captures the essence of the video.

These are multi-step processes for automatically summarizing video content. By leveraging different deep learning models for speech recognition, image captioning, text fusion, and text summarization, the system aims to create comprehensive and informative summaries that capture both the auditory and visual aspects of a video

This paper is structured as follows:SectionIII Proposed methodology.Section IV- Image Captioning. In Section V- Text Fusion. In Section VI -Text Summarization. In SectionVII - Compare the results with other models and finally in SectionVIII- The conclusion.

III. Proposed Methodology

IV. Image Captioning:

Significant progress has been made in computer vision and natural language processing, but many real-world tasks involve multiple data types, like images and text. Multi-modal video summarization aims to bridge this gap by combining these modalities.

This section focuses on the image captioning aspect of video summarization. By extracting keyframes from a video and applying image captioning techniques, we can generate captions that describe the visual content.

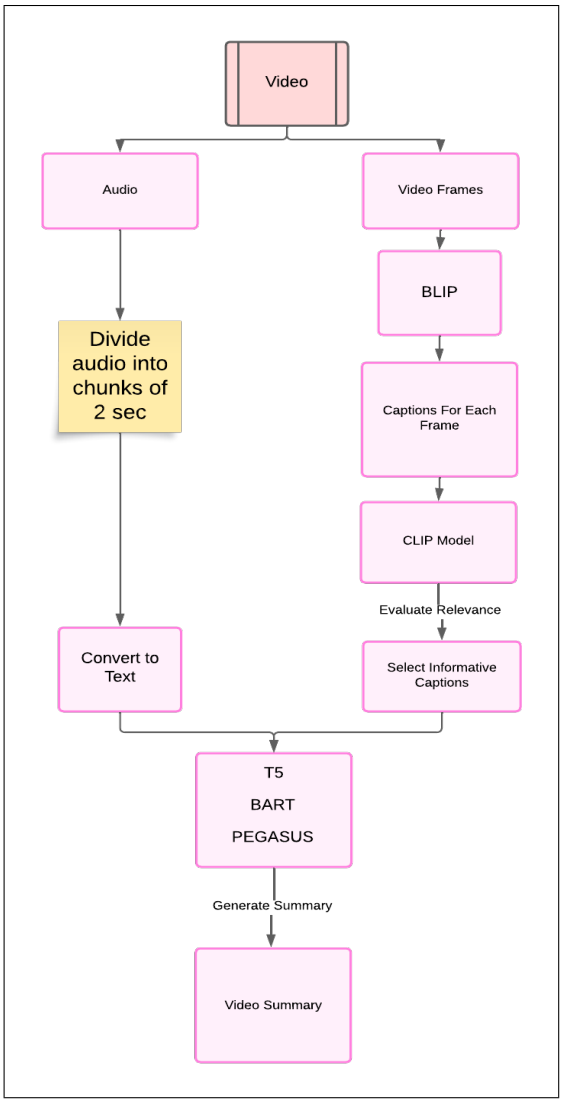

Fig. 1. Proposed Methodology

The provided code utilizes a pre-trained BLIP-2 model for this purpose. BLIP-2 takes an image and text as input, and in our case, it generates captions for each frame. The code then evaluates the similarity between the generated caption and the image itself. Only captions with a high degree of similarity are considered valid, ensuring a degree of accuracy in the descriptions.

These extracted captions, representing the visual content of the video, can then be integrated with the transcribed audio to create a more comprehensive summary. This combined summary leverages both the spoken words and the visual information for a richer understanding of the video content.

This approach showcases the potential of combining computer vision and natural language processing for tasks like video summarization. By utilizing pre-trained models like BLIP-2, we can efficiently extract meaningful descriptions from visual data, contributing to the development of robust multi-modal summarization systems.

BLIP-2 tackles the challenge of combining visual and language information by introducing a lightweight transformer model called Q-Former. This model acts as a bridge between a pre-trained image encoder (frozen and not updated during training) and a large language model (LLM, also frozen). Q-Former is the only trainable component in BLIP-2. It consists of two submodules that share the same processing layers: one for image features and another for text. The image submodule interacts with the frozen image encoder to extract visual information, while the text submodule can handle both encoding and decoding text. Notably, the image submodule extracts a fixed number of features regardless of the image size.

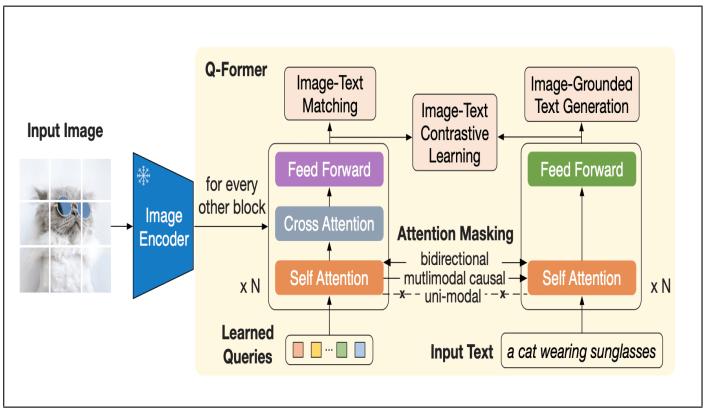

Fig. 2. Q-Former in BLIP

The training for Q-Former happens in two stages. The first stage focuses on building relationships between the image and text information without them directly "seeing" each other. This is achieved through three objectives: maximizing similarity between queries and the most relevant part of the text description, generating text descriptions based on the image with limited interaction between queries and text, and finally, a classification task to determine if a given text description matches the image. During the second stage, the pre-trained queries are used as an informative prefix for the LLM input. This essentially involves training the LLM to generate text descriptions based on the image with guidance from the pre-trained queries. Overall, BLIP-2 demonstrates a flexible approach that allows using any pre-trained image encoder and LLM as long as the Q-Former model is trained for the specific combination.

V. Evaluation of Captions using CLIP

Our model represents an enhanced iteration of the base CLIP model outlined previously. It operates by ingesting two primary inputs: a batch comprising captions and another batch containing images. These inputs undergo individual processing, with captions passing through the CLIP text encoder and images through the image encoder within the model architecture. During the training phase, the model leverages contrastive learning techniques to cultivate a cohesive embedding space that unifies image and caption representations. Within this embedding

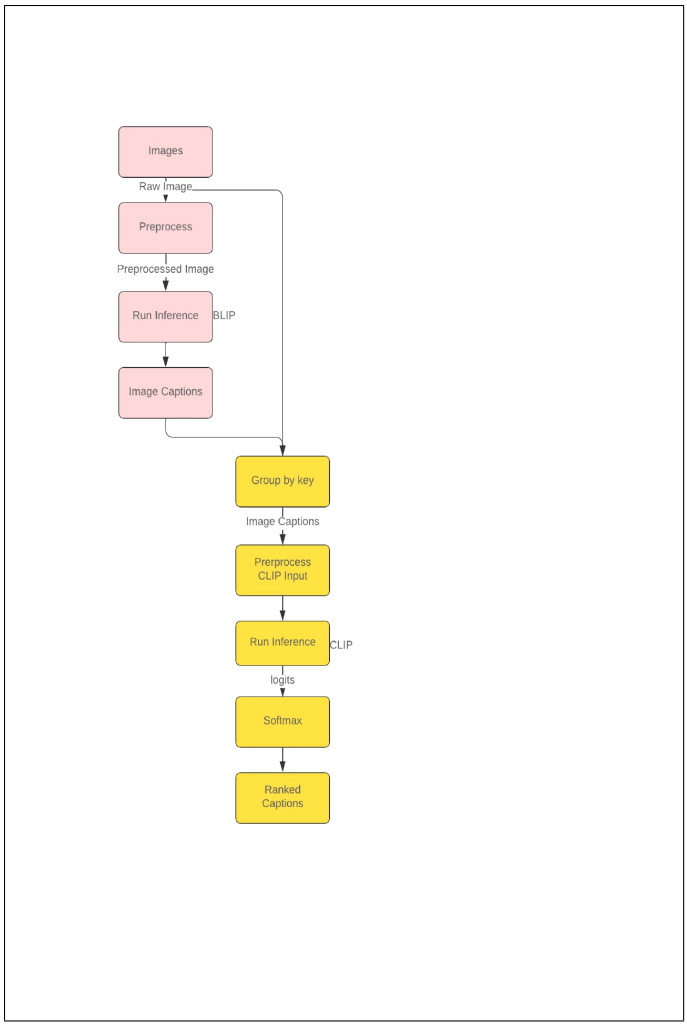

Fig. 3. Caption generation and Evaluation using BLIP-2 and CLIP respectively

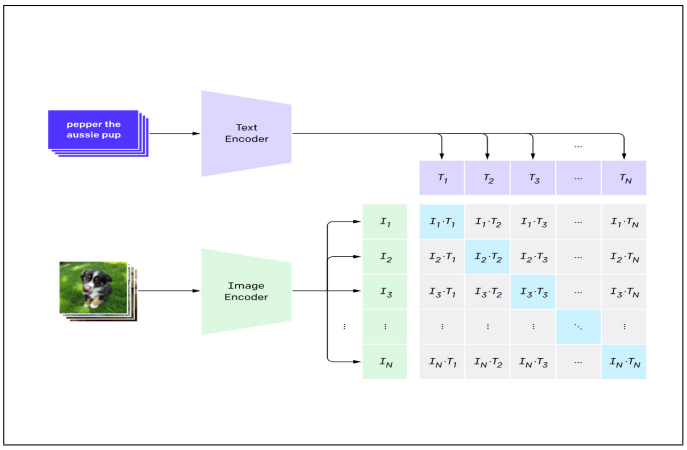

Fig. 4. Constractive pre-training

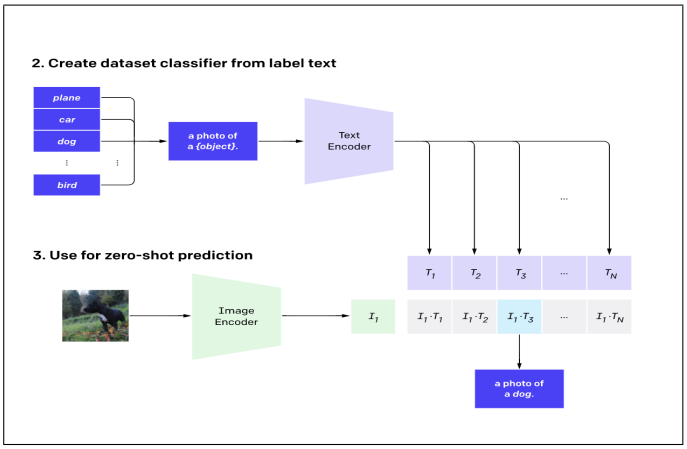

Fig. 5. Creating dataset from label text

space, images and their corresponding captions are intuitively drawn together, fostering proximity. Similarly, akin images and captions converge closely, reinforcing semantic similarities. Conversely, instances of dissimilar images and captions or those belonging to disparate images are naturally pushed farther apart, delineating distinct semantic boundaries within the embedding space.

The two crucial steps in the image captioning process, leveraging the BLIP and CLIP models. Firstly, the BLIP model generated a set of captions for each image, aiming to encapsulate its essence. Each caption underwent a meticulous generation process, ensuring relevance and coherence. The diversity of captions generated per image allowed for a rich spectrum of interpretations.

Subsequently, the CLIP model played a pivotal role in ranking these generated captions based on their fidelity to the underlying image content. Employing its sophisticated inference mechanism, CLIP assigned probabilities to each caption, reflecting its likelihood of accurately representing the image. This step not only facilitated the identification of top-performing captions but also provided insights into the nuanced relationship between images and their textual descriptions.

The integration of both BLIP and CLIP within the pipeline showcased the synergistic potential of multi-model inference in image captioning tasks. The complementary strengths of these models, from generative prowess to discriminative acumen, underscored the pipeline's efficacy in producing meaningful and contextually relevant captions. If the probability of the caption is less than 0.5 then the caption is not added to the video text list the caption will be removed we have added this because it may cause incorrect or ambiguous data will be added to the video text which can cause how evaluation scores in summarization.

In conclusion, the pipeline demonstrated the seamless orchestration of BLIP and CLIP models to deliver a robust image captioning solution. Through their collaborative efforts, the models showcased their proficiency in capturing the essence of images and articulating them into coherent textual representations.

VI. Fusion of Audio and Video Data

Given one video consists of N frames, the image encoder will first map each frame/image into K_f image

embedding vectors, yielding video frame representations $V = [v1, v2, ..., vN]$ where $v_i \in R^{K_f x d_f}$ is the set of d_f dimensional image embeddings corresponding to the i-th frame. The reason why I adopted this method is that if we normally merge two lists. We only take every 50th frame because for every video frame we need to combine it with audio frame text for every video frame audio frame text can be NULLL or a word and also for every frame description of doesn't get changed we do this so that fusion of the audio data and video data will be meaningful and also for performance

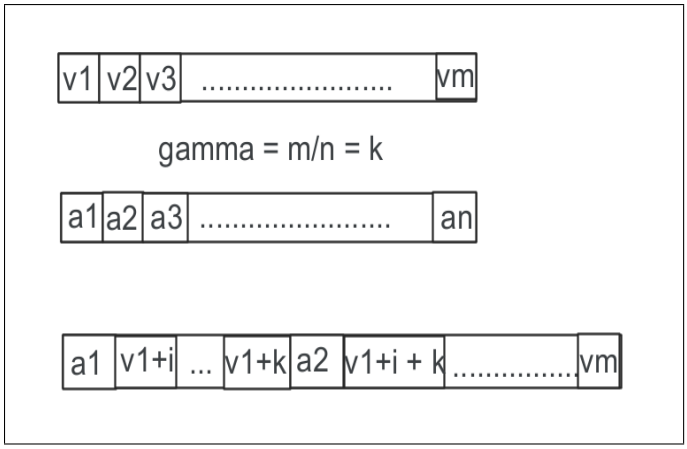

Fig. 6. Merging audio text vector and video text vector in a better way

The process of merging the information from the audio and video goes beyond simple concatenation. Instead, a more sophisticated strategy is employed using a custom function named `merge_lists_and_insert`. This function acts like a conductor, orchestrating the placement of video captions within the audio sentences list. It achieves this by calculating a critical value called Gamma. Gamma isn't just any number; it's a ratio specifically designed to consider the relative amount of information present in each source. Imagine Gamma as a balancing scale, with the number of audio sentences on one side and the number of video captions on the other. The function calculates the value of Gamma by dividing the number of audio sentences by the number of video captions. This ratio provides valuable insight into how much audio information exists compared to the visual content captured in the captions.

For instance, if Gamma is 2, it indicates that there are twice as many audio sentences as video captions. This scenario suggests a higher density of information in the spoken audio compared to the captured visuals. To account for this, the `merge_lists_and_insert` function strategically inserts two audio sentences after each video caption. This ensures a balanced representation of both audio and visual content in the final summary. By strategically placing the video captions throughout the audio sentences based on the calculated Gamma, the function guarantees that the final summary incorporates important details from both the spoken audio and the corresponding visuals from the video, creating a richer and more informative outcome.

VII. Text Summarization

This study compares three pre-trained models, T5, BART, and PEGASUS, for their ability to summarize text. The code first defines functions to load the models and generate summaries with a controllable length. It then applies these functions to a lengthy example text and displays the original text alongside the summaries produced by each model. Finally, the code calculates BLEU, ROUGE, and BERTScore metrics to evaluate the quality of the summaries compared to a human-written reference summary. These metrics provide insights into how well each model captures the essential information from the original text and presents it in a concise and informative way.

BLEU score measures the similarity between the generated summary and a set of reference summaries by counting the number of n-grams (sequences of n words) that they share in common. The ROUGE score is another common metric for text summarization evaluation. It considers not only the n-gram overlap but also recall, which is the proportion of information in the reference summary that is also present in the generated summary. BERTScore is a more recent metric that leverages a pre-trained deep learning model to score the semantic similarity between the generated summary and the original text. By comparing the performance of these three models on the same task, this project provides valuable insights into the strengths and weaknesses of each approach to text summarization.

VIII. Baseline Techniques

In the paper [1] the author has used BLIP-2 and CLIP for Image captioning and evaluation respectively. Audio is divided into n partitions and video for every frame caption is generated we take both of them and pass them through the LLM model for Q/A based on a video. But in our study, we are going not going to divide audio into n partitions because it won't create meaningful data because audio at a frame length might be a word or a letter which when combined with video data doesn't create a meaning full data. So, we are going to divide audio data when there is comma(,) or a fullstop(.) then it can crate meaningful sentences which when combined with video data create a meaningful multi-model text which is useful fro summarization

IX. Experimental Analysis

Evaluation Metrics:

- **BLEU Score:** This metric measures the similarity between the generated summary and a set of reference summaries by counting n-gram matches (sequences of n words). The provided code reports very low BLEU scores for all three models, which is likely because the reference summaries and the generated summaries don't have many exact word matches. However, the

BLEU score is often criticized for not always reflecting human judgments of summary quality.

- **ROUGE Score:** This metric considers not only n-gram overlap but also recall, which is the proportion of information in the reference summary that is also present in the generated summary. The code shows that PEGASUS achieves the highest ROUGE scores, followed by T5 and BART. This suggests that PEGASUS summaries capture the essential information from the original text more effectively than the other two models.
- **BERTScore:** This metric leverages a pre-trained deep learning model to score the semantic similarity between the generated summary and the original text. The results indicate that T5 has the highest BERTScore F1, followed by BART and PEGASUS. This suggests that T5 summaries may be the most semantically similar to the original text.

X. Comparing the results of T5, BART, PEGASUS with BLEU, ROUGE and BERTScores

Metric	T5	BART	PEGASUS
BLEU	1.25e-231	1.25e-231	1.23e-231
ROUGE-1	0.2857	0.2143	0.1429
ROUGE-2	0.1429	0.1429	0.0000
ROUGE-L	0.2857	0.2143	0.1429
BERTScore-P	0.8870	0.8891	0.8705
BERTScore-R	0.8735	0.8683	0.8479
BERTScore-F1	0.8802	0.8786	0.8590

Overall Analysis: While BLEU scores are low for all models, ROUGE and BERTScore metrics provide more insights. PEGASUS achieves the best ROUGE scores, indicating it captures the key information from the text well. However, T5 has the highest BERTScore F1, suggesting its summaries might be more semantically similar to the original text. It's important to consider the trade-off between factual correctness (ROUGE) and semantic similarity (BERTScore) when choosing the best model for your summarization task.

XI. Conclusion

This project successfully developed a multi-model video summarization system that combines the strengths of automatic speech recognition and image captioning to create summaries that are more informative and engaging than traditional audio-only methods. The system first employs Whisper, a state-of-the-art speech recognition model, to transcribe the audio content of a YouTube video. Next, it extracts key frames at regular intervals and utilizes blip-image-captioning-base, a pre-trained image captioning model, to generate captions for these frames. To ensure the captions accurately reflect the corresponding audio segments, the system leverages CLIP, a powerful image-text similarity scoring model. CLIP assigns a confidence score to each caption, indicating how well it aligns with the transcribed audio. By incorporating these confidence scores, the system can effectively select the most relevant captions to complement the transcribed sentences. Finally, the system merges the informative audio segments with the most relevant image captions, resulting in a multi-model summary that captures both the auditory and visual aspects of the video content. This comprehensive approach provides viewers with a richer understanding of the video, making it an ideal tool for applications such as video search, video browsing, and educational content creation.

References

[1] H. Zhang, X. Li, and L. Bing, "Video-llama: An instruction-tuned audio-visual language model for video understanding," *ArXiv*, vol. abs/2306.02858, 2023.

[2] J. Li, D. Li, S. Savarese, and S. C. H. Hoi, "Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models," in *International Conference on Machine Learning*, 2023.

[3] S. Agarwal, G. Krueger, J. Clark, A. Radford, J. W. Kim, and M. Brundage, "Evaluating clip: Towards characterization of broader capabilities and downstream implications," *ArXiv*, vol. abs/2108.02818, 2021.

[4] S. Shen, L. H. Li, H. Tan, M. Bansal, A. Rohrbach, K.-W. Chang, Z. Yao, and K. Keutzer, "How much can clip benefit vision-and-language tasks?," *arXiv preprint arXiv:2107.06383*, 2021.

[5] J. Li, D. Li, C. Xiong, and S. Hoi, "BLIP: Bootstrapping language-image pre-training for unified vision-language understanding and generation," in *Proceedings of the 39th International Conference on Machine Learning* (K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, eds.), vol. 162 of *Proceedings of Machine Learning Research*, pp. 12888–12900, PMLR, 17–23 Jul 2022.

[6] H. Chefer, S. Benaim, R. Paiss, and L. Wolf, "Image-based clip-guided essence transfer," in *European Conference on Computer Vision*, pp. 695–711, Springer, 2022.

[7] A. G. Etemad, A. I. Abidi, and M. Chhabra, "Fine-tuned t5 for abstractive summarization," *International Journal of Performability Engineering*, vol. 17, no. 10, p. 900, 2021.

[8] S. Shleifer and A. M. Rush, "Pre-trained summarization distillation," *arXiv preprint arXiv:2010.13002*, 2020.

[9] J. Zhang, Y. Zhao, M. Saleh, and P. Liu, "Pegasus: Pre-training with extracted gap-sentences for abstractive summarization," in *International conference on machine learning*, pp. 11328–11339, PMLR, 2020.

[10] W. Kryściński, R. Paulus, C. Xiong, and R. Socher, "Improving abstraction in text summarization," *arXiv preprint arXiv:1808.07913*, 2018.

[11] J. Steinberger and K. Jezek, "Evaluation measures for text summarization," *Computing and Informatics*, vol. 28, no. 2, p. 251, 2009.

[1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11]