Text Summarization of Educational Videos

Jose Carlos Salinas
Facultad de Ingeniería
Universidad del Pacífico
Lima, Perú
jc.salinasm@alum.up.edu.pe

César Cabezas
Facultad de Ingeniería
Universidad del Pacífico
Lima, Perú
cm.cabezasg@alum.up.edu.pe

Franz Figueroa
Facultad de Ingeniería
Universidad del Pacífico
Lima, Perú
f.figueroag@alum.up.edu.pe

***Abstract*—The COVID-19 pandemic significantly accelerated the adoption of digital educational content, resulting in an unprecedented increase in recorded audiovisual material. While this content ensures continuity in education, it also presents challenges, particularly in navigating and extracting relevant information from lengthy videos. This study proposes a multimodal transformer-based model designed to generate text summaries of educational videos by integrating video, audio, and textual data. The model utilizes advanced techniques such as frame sampling, positional encoding, and a specialized dual-decoder architecture to address the diverse needs of free and paid summarization tasks. Evaluation on a dataset of 1029 educational videos shows that the paid summaries outperform their free counterparts across BLEU and ROUGE metrics. Despite these advances, limitations related to computational resources and dataset diversity were identified. Future work should focus on incorporating more diverse educational videos to improve the model's generalization and utility in real-world applications.**

***Index Terms*—Education, Video summarization, Multimodal, Transformers, BLEU, ROUGE.**

I. Introduction

The COVID-19 pandemic caused an unprecedented global disruption in the educational sector, forcing over 190 countries to close educational institutions and affecting more than 1.6 billion students, according to UNESCO [1]. This event triggered a rapid transition to distance learning, accelerating the use of digital platforms as a means to continue classes and reduce learning disruption [2]. In response, universities, schools, and other educational institutions began generating a significant amount of recorded audiovisual content, ranging from full courses to specialized lectures across various disciplines [3]. This explosion of content has been critical in ensuring the continuity of education, even in a context where in-person learning was severely restricted.

However, this vast access to educational audiovisual content has introduced new challenges. One of the most prominent problems, according to Bates [4], lies not just in the availability of such material but in the difficulties students face when navigating and extracting relevant information from long videos. Unlike written content, where keyword searches are possible, videos often do not provide effective mechanisms for quickly accessing specific sections of interest, making the process of finding precise information tedious [5]. This issue is particularly pronounced in lengthy videos, such as lectures or seminars, where locating a key explanation or concept can require significant time investment from students [6]. In higher education settings, where students are expected to master large volumes of information within tight deadlines, this challenge can negatively impact the efficiency of the learning process.

Moreover, with the growing volume of audiovisual content, both students and educators face the difficulty of managing this vast array of resources effectively. Many students find themselves overwhelmed when attempting to review and process large amounts of video content in preparation for exams or assignments, which reduces the overall learning efficiency and increases cognitive load [7]. At the same time, educators face the challenge of ensuring that their recorded lectures are accessible and navigable, particularly in contexts where students have different learning styles and technological capabilities.

A survey conducted by our research group of 72 students at Universidad del Pacífico revealed that 68.1% of respondents use the university's educational platform to search for recorded lectures, and 22.4% failed to find the topic they were looking for in the available videos. In addition, 79.6% of respondents felt that videos on the platform were poorly labeled, and 94.4% of those using other platforms such as YouTube, Udemy or Coursera also reported similar labeling problems. These findings evidence an urgent need to improve the organization and accessibility of content in educational videos, both in our institution and in the educational field in general.

In this context, automatic summarization of educational videos emerges as a promising solution to mitigate these challenges. Automatic summaries could help students quickly identify the main topics of a video, thus facilitating access to essential information and optimizing the learning process [8]. By reducing the time spent searching for relevant segments, these summaries allow students to focus more on the key content and improve their overall understanding of the material. This would not only benefit students in terms of accessibility but also ease the burden on educators, who could offer more manageable content focused on the most important aspects.

In the Spanish-speaking context, where the amount of educational content available in Spanish is increasing significantly, the implementation of automatic summarization tools becomes even more relevant. In many Latin American countries and Spain, access to quality education through digital platforms has grown exponentially in recent years, driven in part by the pandemic [9]. However, the lack of tools that allow for efficient navigation within these resources remains a considerable barrier. The ability to automate the creation of summaries in educational videos would not only optimize the use of these materials but also promote greater equity in access to education, especially in regions where connectivity is limited and students rely on time optimization for their studies [10].

Given this, the incorporation of artificial intelligence techniques for the creation of educational video summaries represents a unique opportunity to transform the way students interact with digital resources. By providing tools that enable them to navigate more efficiently through the vast ocean of available information, accessibility is improved, and a more autonomous and effective learning process is facilitated.

II. Literature Review

The task of video summarization has evolved significantly, incorporating multiple deep learning techniques. While transformers have gained popularity in generating meaningful summaries from video content, other methods such as Graph Convolutional Networks (GCNs) have also demonstrated their potential in handling spatial and temporal relationships. In this section, we will review some of the most relevant works in the field, highlighting their methodologies, datasets, and results, and connecting them to the present study, which focuses on generating text-based summaries from both the audio and visual frames of educational videos.

One of the foundational works in this area is VideoBERT: A Joint Model for Video and Language Representation Learning by Sun et al. [11]. This work addresses the challenge of learning joint representations of video and language without requiring explicit supervision. VideoBERT builds upon the BERT architecture, a transformer model originally designed for text-based tasks, and adapts it to process video frames and their corresponding transcriptions. The model uses Automatic Speech Recognition (ASR) to extract textual data from video speech and employs vector quantization to represent the visual data. VideoBERT was trained on the YouCook II dataset, which consists of instructional cooking videos. The model achieves state-of-the-art performance in video captioning and action classification, demonstrating its ability to capture high-level semantic features from both video and language. This capability is essential for the current project, where both audio and visual information need to be summarized in a cohesive text format.

Similarly, Wei et al.'s work, Video Summarization via Semantic Attended Networks [12], proposes a supervised learning framework for selecting the most semantically relevant video frames and generating a coherent summary. The authors introduce a frame selector that identifies key frames based on their semantic importance, and an LSTM-based video descriptor to ensure that the selected frames maintain continuity across the summary. This model was evaluated on the SumMe and TVSum datasets, showing its capability to capture the most informative aspects of a video while reducing it to a compact, meaningful summary. This approach is highly relevant for summarizing educational videos, where key moments such as explanations of difficult concepts or demonstrations are critical for generating accurate summaries.

The approach of Mahasseni et al., in their paper Unsupervised Video Summarization with Adversarial Networks [13], presents a unique method for video summarization through unsupervised learning. The authors propose an adversarial network where a generative network produces video summaries, and a discriminative network evaluates the quality of these summaries by comparing them to human-generated ones. This setup helps improve the quality of the generated summaries by pushing the model to create more realistic outputs. The model utilizes LSTM networks for both the generator and the discriminator, focusing on capturing the temporal structure of the video. Tested on the SumMe and TVSum datasets, this model demonstrated its ability to create realistic and concise summaries even without labeled data, making it particularly useful when labeled data for educational videos is scarce.

Zhou et al., in their work End-to-End Dense Video Captioning with Masked Transformer [14], introduce a masked transformer architecture for generating dense captions that describe video content in detail. Their model trains by masking portions of the input video, forcing it to learn contextual relationships between the visible parts, which results in detailed and contextually relevant captions. The model was evaluated on the ActivityNet Captions dataset, where it outperformed previous methods. The masked transformer architecture is especially relevant for this project, as it can handle long-range dependencies in educational videos, generating coherent text summaries that reflect both audio and visual content.

In a different approach, Li et al. propose the use of Graph Convolutional Networks (GCNs) in their work Graph Convolutional Transformer for Video Summarization [15]. The authors introduce a hybrid model that applies GCNs to capture spatial relationships between video frames and transformers to model the temporal dependencies. This method is particularly effective for summarizing videos with complex interactions between objects across time, such as instructional or educational videos. The model was tested on the ActivityNet and YouTube-8M datasets, showing improved performance in video summarization tasks. While the current project focuses on transformers, the application of GCNs demonstrates the potential of exploring spatial relationships between video frames to enhance summary quality, especially when combined with temporal information.

Finally, Narayan et al.'s paper Multimodal Transformer for Video Summarization [16] highlights the importance of incorporating both audio and visual modalities for effective video summarization. Their model processes audio and visual streams separately using transformers and then combines these streams through a cross-attention mechanism. The model was evaluated on the How2 and YouCook II datasets, achieving state-of-the-art results in summarization. This approach aligns closely with the goals of the present study, which also seeks to combine audio and visual data to generate comprehensive text summaries from educational videos. The use of both audio and frames provides a more complete representation of the video's content, ensuring that important information from both modalities is included in the final summary.

In conclusion, the reviewed works demonstrate a range of methodologies for video summarization, from supervised approaches focusing on semantic relevance to unsupervised methods utilizing adversarial learning. The use of transformers and GCNs for modeling spatial and temporal relationships in videos has shown significant promise. For this project, the integration of both audio and video frames using a multimodal approach, as highlighted by Narayan et al., provides a compelling strategy for generating accurate and informative text summaries from educational videos.

III. Theoretical Framework

A. MoviePy

MoviePy is a versatile and open-source Python library designed for video editing. [17] MoviePy supports various video formats and allows for complex editing tasks, such as cutting, concatenating, applying effects, and manipulating audio tracks. One of its key features is its ease of use, as it allows developers and users to handle video processing tasks with just a few lines of Python code.

B. Speech_recognition

The SpeechRecognition library in Python provides a convenient interface for interacting with various speech-to-text APIs, including the Google Web Speech API [18]. The library's recognize_google() method interacts with the Google Web Speech API, where the audio data is processed by Google's machine learning models to identify spoken words and return them as text. This method supports multiple languages and can handle both online and offline audio files. One of the key advantages is that the Google Web Speech API operates in the cloud, allowing it to leverage advanced machine learning models for speech-to-text conversion without requiring heavy local computation.

C. BERT2BERT Shared Spanish model

The BERT2BERT Shared Spanish model is a fine-tuned version of the BERT model specifically adapted for text summarization tasks in Spanish developed by mrm8488 and hosted on Hugging Face [19]. this model leverages the BERT architecture in both the encoder and decoder, following the ”BERT2BERT” setup. This approach builds on the strengths of BERT's bidirectional context understanding for both encoding and generating summaries, fine-tuning it on a Spanish corpus to generate high-quality text summaries. The model is particularly effective for summarizing large bodies of text into shorter, coherent summaries while maintaining key information. By sharing weights between the encoder and decoder, it optimizes the training process and improves performance.

D. Positional Encoding

Positional Encoding is a fundamental technique in Transformer models that allows word order information to be incorporated into a sequence. Since Transformers process inputs simultaneously rather than sequentially, they lack an inherent structure to capture word order. Positional Encoding solves this problem by assigning a unique representation to each position in the sequence, making it easier for the model to distinguish between tokens based on their position [20]. This encoding is commonly implemented using sine and cosine functions of different frequencies to generate unique vectors for each position. This mathematical approach allows the model to effectively distinguish between near and distant positions, while maintaining context and word relationships. In the context of generating summaries of educational videos, Positional Encoding is essential for the model to understand the temporal sequence of frames and audio. The equations are given below:

$$PE_{(pos,2i)} = \sin\left(\frac{pos}{10000^{\frac{2i}{d_{model}}}}\right)$$

$$PE_{(pos,2i+1)} = \cos\left(\frac{pos}{10000^{\frac{2i}{d_{model}}}}\right)$$

E. Multi-Head Attention

The *Attention* mechanism, introduced by Vaswani et al. in their influential paper “Attention is All You Need” [26], fundamentally transformed the approach to processing sequential data in neural networks. Unlike traditional recurrent architectures such as LSTM or GRU, which process sequences sequentially and can struggle with long-term dependencies, the Attention mechanism allows models to weigh the relevance of different parts of the input data dynamically. This capability enhances the model's ability to capture contextual relationships within the data effectively.

Multi-Head Attention is an extension of the basic Attention mechanism that enables the model to focus on multiple aspects of the input simultaneously. This is achieved by projecting the input vectors into multiple subspaces, each corresponding to a distinct ”head.” Specifically, the process involves the following steps:

1) **Linear Projections:** The input vectors are linearly transformed into three separate sets of vectors: *queries* (Q), *keys* (K), and *values* (V). These projections are performed using distinct weight matrices for each head.
2) **Parallel Attention Computation:** Each head independently performs the scaled dot-product Attention operation:

   $$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V$$

   where d_k is the dimensionality of the keys. This step allows each head to focus on different parts of the input sequence.
3) **Concatenation and Final Projection:** The outputs from all heads are concatenated and then linearly transformed into a single output vector. This aggregation integrates the diverse contextual information captured by each head.

The number of heads (h) is a hyperparameter that determines how many parallel Attention mechanisms are employed.

In the original Transformer model, $h = 8$ was chosen, allowing the model to capture a variety of relationships and dependencies within the data. Each head operates in a lower-dimensional subspace ($d_k = \frac{d_{\text{model}}}{h}$), ensuring that the overall computational complexity remains manageable.

Multi-Head Attention serves two primary roles within the Transformer architecture:

- **Self-Attention:** Within the encoder and decoder layers, Self-Attention allows each position in the sequence to attend to all other positions, facilitating the capture of global dependencies and contextual relationships.
- **Encoder-Decoder Attention:** In the decoder, this mechanism enables the model to focus on relevant parts of the encoder's output when generating each element of the output sequence, ensuring that the generated content is contextually grounded in the input data.

The integration of multiple attention heads enhances the model's ability to understand and represent complex patterns in the data by simultaneously attending to information from different representation subspaces. This multi-faceted approach not only improves the richness of the learned representations but also contributes to the overall efficiency and scalability of the model.

F. Latin hypercube sampling

Latin Hypercube Sampling is a statistical sampling technique that seeks to improve the representativeness of samples drawn from multidimensional distributions. It divides each dimension of the input space into equiprobable intervals, ensuring that each interval is represented at least once in the final sample [27]. This contrasts with pure random sampling, where there is no guarantee that samples will uniformly cover the space. A key feature is that it generates samples in a stratified manner improving the coverage of the variable space.

G. YAMNET

YAMNet is a pre-trained convolutional neural network model designed for audio classification tasks such as sound event recognition and acoustic pattern detection [28]. Based on the MobileNet architecture, YAMNet uses lightweight and efficient convolutional layers that make it suitable for devices with limited computational resources. This model leverages transfer learning, as it has been pre-trained on Google's AudioSet dataset, which contains over 600 categories of sound events, allowing it to learn rich and generalizable representations. YAMNet processes input data in the form of log-Mel spectrograms, a representation that transforms audio signals into a format that highlights the most relevant acoustic features for classification.

IV. Methodology

The proposed methodology describes the design and implementation of a multimodal model designed to generate summaries from heterogeneous data (video, audio and text). This process includes phases of data preparation and processing, design of the transformer-based model, and configuration of the training and inference processes. Each of the key stages is described in detail below. The full metodology is summarized in Figure 1.

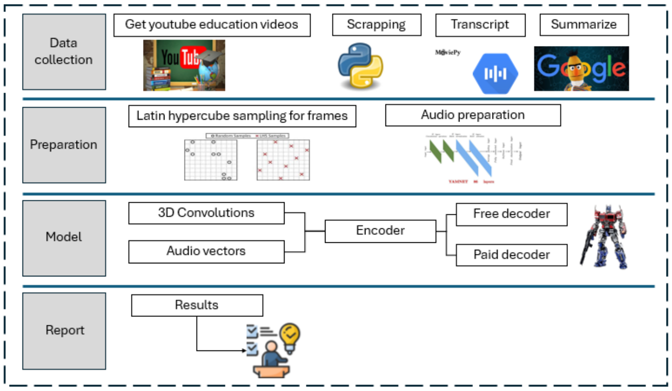

Fig. 1. Diagram of the Methodology

A. Dataset Creation

1) Video and Audio Extraction:

a) Video Extraction: Videos are sourced from platforms such as YouTube, focusing on educational content. Each video is downloaded in its original MP4 format. Before processing, all videos are standardized to a resolution of 480p (854x480 pixels) and a frame rate of 30 FPS using *FFmpeg*. This standardization ensures uniformity in visual quality and frame consistency across the dataset. The audio is resampled to 16 kHz for compatibility with the subsequent audio processing stages.

b) Audio Extraction: Using the *moviepy* library, audio is extracted from each video and converted into WAV format. Audio is split into segments of 30 to 60 seconds to optimize memory usage and improve accuracy in downstream tasks such as transcription.

2) Transcription Process:

a) Transcription: Google's Speech-to-Text API is used to transcribe the extracted audio into text form. This transcription captures the spoken content of the video, which will be used as input for text summarization.

b) Summarization: The transcribed text is summarized using the *BETO* model, a transformer pre-trained in Spanish. This model generates concise and coherent summaries by identifying key information from the transcription.

B. Dataset description

The dataset we are using consists of 1029 videos scraped from YouTube, each paired with its corresponding text summary. These videos cover 9 different categories related to elementary school courses, providing a diverse range of educational material for training the model. The courses are: math, chemistry, biology, religion, mathematical reasoning, verbal reasoning, spanish, history and physics.

In order to explore and identify patterns, we look for the statistical metrics of both videos and audios for further preprocessing. In the case of the videos, we look for the duration in seconds, the width and the height in pixels. The metrics of the videos and the audios are shown in the tables II **??** respectively.

TABLE I
STATISTICAL METRICS OF THE VIDEOS

Metric	Height (pixels)	Width (pixels)	Duration (seconds)
Min.	288	202	8.45
Max.	644	640	2632.12
Mean	385	590	317.39
Median	360	640	225.39
Std. Dev.	79.98	115.27	328.35

TABLE II
STATISTICAL METRICS OF THE AUDIOS

Metric	TOtal of samples per audio	Duration (seconds)
Min.	372557	8.45
Max.	116076544	2632.12
Mean	13997082.47	317.39
Median	9939456	225.39
Std. Dev.	14480365.04	328.35

C. Multimodal Data Preparation

The process begins with data preparation, ensuring consistency and correspondence between video, audio and text modalities.

a) Video preprocessing: Each video is loaded as a tensor in NumPy format with an initial shape of $(frames, height, width, channels)$. It is adjusted to a standard size of 100 frames by applying Latin hyper cube sampling, ensuring that each tensor has the shape $(100, height, width, 3)$. Individual frames are resized to 224 pixels using bilinear interpolation to ensure spatial uniformity.

b) Audio preprocessing: We used the Librosa library from Python in order to load the audios. We normalized them by converting to a tensor and type float32 using tensorflow. Finally, we use the YAMNet to obtain characteristic vectors for each audio, resulting in fixed-length vectors of 1024 dimensions.

c) Text preprocessing: Textual summaries are extracted from files in text format `.txt` and cleaned to remove empty entries. This text will later be used for tokenization.

D. Tokenization

Once the data is prepared, it is integrated into a `tf.data.Dataset` that combines the modalities and transforms them into a format suitable for the model.

a) Efficient loading of modalities: Videos are loaded as shape tensors $(100, 224, 224, 3)$. Audios are processed as feature vectors with fixed length of 1024. Textual summaries are processed as text strings for further tokenization.

b) Text tokenization: The BETO model (BERT in Spanish) is used to tokenize the summaries. This process converts the text into token sequences compatible with the pre-trained BETO vocabulary. Two types of sequences are generated:

- **Sequences for the free decoder:** Limited to 60 tokens ($max\_len\_free = 60$).
- **Sequences for the paid decoder:** Extended up to 1000 tokens ($max\_len\_paid = 1000$).

E. Multimodal Model Design

The proposed model is based on a multimodal architecture with transformers, which includes an encoder and two specialized decoders. Its architecture is designed to integrate video, audio and text features effectively.

a) Video processing: Videos are processed through a stack of three-dimensional convolutional layers. (`Conv3D`) that extract spatial and temporal features. Each layer includes a *pooling* which progressively reduces the spatial dimensions. The final output is projected to a latent space with fixed dimensionality. (d_model) to be compatible with the encoder.

b) Audio processing: The audios are processed with a `LSTM (Long Short-Term Memory)`, which captures the temporal relationships within the feature vectors. The output of the LSTM is projected to the same latent space. (d_model) for integration with video features.

c) Fusion of modalities and encoder: The video and audio features are combined by a summing operation and enriched with *Positional Encoding*. The combined representation is processed by a transformer-based encoder that includes multiple blocks. Each block contains:

- A *Multihead Attention* layer, that models complex relationships between features.
- A *feed-forward* network to capture nonlinear patterns.
- Layer normalization and *dropout* to improve training stability and prevent overfitting.

d) Specialized decoders:

- **Free decoder:** Designed to generate short summaries. It uses only the processed video features and passes through two simplified transformer blocks.
- **Paid decoder:** Optimized for long summaries, it uses the combined video and audio features. This decoder includes five complete transformer blocks.

Both decoders produce token sequences that are transformed into text using 66the BETO tokenizer.

The proposed architecture is based on 3 blocks that are repeated throughout the layers, which are shown in Figure 2.

The complete architecture is shown in Figure 3, making use of convolution, pooling, dense, normalization and embedding layers.

F. Inference and Validation

The inference process allows to evaluate the sample-by-sample model and decode the predictions to obtain readable text.

a) Decoding predictions: Model predictions (probability vectors) are transformed into token sequences using the BETO tokenizer. Decoding stops when the completion token is encountered to avoid redundant text.

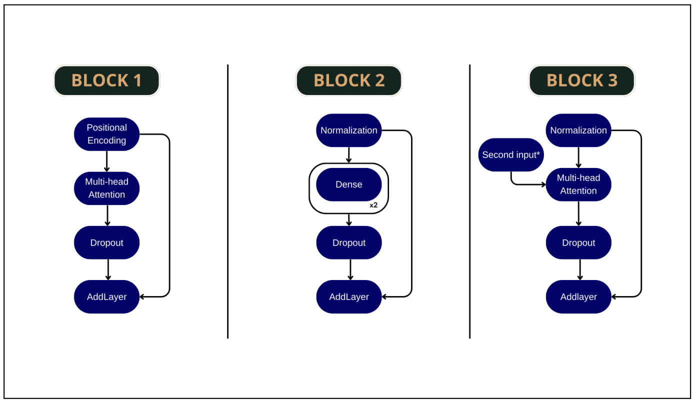

Fig. 2. Blocks used in the architecture

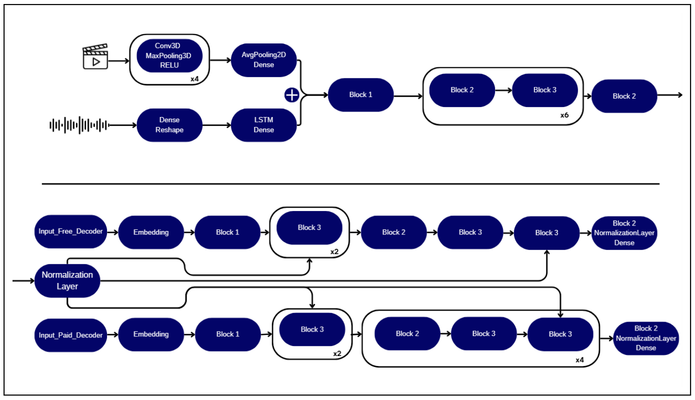

Fig. 3. Architecture

b) Comparison with original texts: The predictions generated are compared with the original texts to validate the quality and accuracy of the summaries in both scenarios.

V. Evaluation

The generated summaries are evaluated using the following metrics:

- **ROUGE (Recall-Oriented Understudy for Gisting Evaluation):** Measures the overlap between the generated summary and the reference summary. It includes several variants:
  - **ROUGE-1:** Evaluates unigram (single word) overlap.
  - **ROUGE-2:** Assesses bigram (two-word sequence) overlap.
  - **ROUGE-L:** Measures the longest common subsequence, capturing sentence-level structure similarity.
- **BLEU (Bilingual Evaluation Understudy):** Evaluates the fluency and accuracy of the generated text by assessing how closely it matches the reference summaries through precision of n-gram matches.

VI. Results

In this section, we present the average evaluation metrics for both free and paid summaries generated by our model. The metrics include BLEU, ROUGE-1, ROUGE-2, and ROUGE-L scores.

TABLE III
Average Metrics for Free and Paid Summaries

Metric	Free Summaries	Paid Summaries
BLEU	44.12	54.51
ROUGE-1	0.70	0.77
ROUGE-2	0.51	0.61
ROUGE-L	0.69	0.76

As illustrated in Table III, the paid summaries demonstrate superior performance compared to the free summaries across all evaluated metrics. Specifically, the BLEU score for paid summaries is 54.51, significantly higher than the 44.12 observed for free summaries. Similarly, ROUGE-1, ROUGE-2, and ROUGE-L scores are consistently higher for paid summaries, indicating a greater overlap with reference summaries in terms of unigram, bigram, and longest common subsequence matches, respectively.

VII. Discussion

The results obtained reflect the performance of the proposed multimodal model in the generation of summaries for educational videos, showing both its inherent strengths and limitations. This section addresses the critical analysis of the findings, as well as the implications of the methodological decisions and the areas for improvement identified.

The analysis of BLEU and ROUGE metrics reveals a notably superior performance in paid summaries compared to free ones. This performance is due to the greater ability of the specialized long summary decoder to capture more complex and detailed features of the processed videos and audios. For example, the BLEU score of 54.51 for paid summaries indicates higher alignment with the reference summaries, while the ROUGE-1, ROUGE-2 and ROUGE-L scores demonstrate higher robustness in terms of n-gram matches and overall structure.

However, it is important to note that while the metrics are consistently higher for paid summaries, the differences between the two categories suggest that free summaries also capture key information efficiently, albeit in less depth. This highlights the effectiveness of using a multimodal approach even with simplified decoders.

Despite the advances and optimizations made in the model, this work faces several limitations inherent to the design and available resources, which could impact both the accuracy of the results and their applicability to more diverse scenarios.

First, due to computational resource constraints, it was necessary to apply frame sampling to the videos to reduce the dimensionality of the inputs. This implied limiting each video to 100 representative frames, which could lead to the loss of critical information, especially in longer videos where temporal compression may not capture all relevant events. Likewise, the dimensions of the resulting vectors and matrices were also intentionally limited, both for video (224×224 pixels per frame) and audio (1024 features). These constraints ensure that the model can be efficiently trained and evaluated within the available GPU memory limits, but may compromise the model's ability to process more detailed inputs.

Another important aspect is the duration of the videos included in the dataset. While the videos used have relatively uniform and manageable lengths, the model may face problems when processing significantly longer videos. For example, a long video would require a more sophisticated sampling or segmentation strategy to ensure that all relevant information is captured without exceeding the model's input constraints. This limitation could negatively impact the generalization of the model to domains where videos tend to be longer, such as lectures or recordings of entire classes.

In addition, the videos used to train the model belong to the educational domain, but do not necessarily represent structured classes such as those taught in a school or university setting. The educational videos used are usually short, with well-defined narratives and carefully selected content. However, in the case of introducing videos of actual lectures, which tend to be longer, less structured, and with more noise (interactions with students, pauses, and deviations from the topic), the model may fail to generate coherent and representative summaries. This suggests that an extension of the work should consider including more diverse data, specifically from real classes, to improve the robustness of the system in such scenarios.

Finally, although the model is effective in controlled scenarios, these limitations highlight the need to optimize the pipeline to handle higher computational resources, explore strategies for processing longer duration videos, and expand the dataset with examples more representative of the real educational environment. These steps would improve both the generalizability and utility of the model in broader practical applications.

VIII. Conclusion

The results of this work highlight the potential of transformer-based multimodal models to generate effective summaries of educational videos. Paid summaries demonstrated superior performance on metrics such as BLEU and ROUGE, evidencing their ability to capture more detailed and structured information. This suggests that the use of specialized decoders is crucial to address tasks of higher complexity in the educational domain.

However, important limitations were identified, mainly related to computational resources and representativeness of the dataset. The need to reduce dimensionality by sampling frames and limiting the dimensions of the inputs could compromise the model's ability to generalize to longer videos or videos with more heterogeneous content. In addition, the structured nature of the videos used does not fully reflect the challenges present in real classrooms, where the data tend to be messier and noisier.

For future work, it is proposed to incorporate more advanced sampling strategies, such as adaptive sampling, and to extend the dataset with full class videos to improve the robustness of the model. It would also be beneficial to optimize the pipeline to handle greater computational resources, which would allow processing more complex and diversified inputs.

In conclusion, although the model has proven to be effective in controlled scenarios, its implementation in real educational environments will require significant improvements in data preparation and architecture to maximize its applicability and utility.

References

[1] UNESCO, *COVID-19 Educational Disruption and Response*, 2020. Available: https://en.unesco.org/news/covid-19-educational-disruption-and-response

[2] C. Hodges, S. Moore, B. Lockee, T. Trust, and A. Bond, *The Difference Between Emergency Remote Teaching and Online Learning*, Educause Review, 2020. Available: https://er.educause.edu/articles/2020/3/the-difference-between-emergency-remote-teaching-and-online-learning

[3] N. Selwyn, *Education and Technology: Key Issues and Debates*, 2nd ed., Bloomsbury Academic, 2020.

[4] A. Bates, *Online Learning and Distance Education Resources*, 2021. Available: https://www.tonybates.ca

[5] M. Pérez-López, *Navigating Educational Videos: Problems and Solutions*, Revista de Educación y Tecnología, vol. 12, pp. 23–45, 2020.

[6] P. Guo, J. Kim, and R. Rubin, *How Video Production Affects Student Engagement: An Empirical Study of MOOC Videos*, Proceedings of the First ACM Conference on Learning at Scale, pp. 41–50, 2014.

[7] A. Smith and J. Jones, *Cognitive Load in Online Learning: Challenges and Solutions*, Journal of Online Learning Research, vol. 5, pp. 120–140, 2021.

[8] L. López, *Automatic Summarization of Educational Videos: A Promising Solution*, International Journal of Educational Technology, vol. 18, pp. 33–47, 2021.

[9] M. Gómez and S. Martínez, *The Impact of Online Learning in Latin America During the Pandemic*, Revista Latinoamericana de Educación, vol. 25, pp. 15–28, 2021.

[10] R. Torres and L. Fernández, *Optimizing Study Time Through Technological Tools*, Education and Technology in the Digital Age, vol. 3, pp. 97–114, 2021.

[11] C. Sun, A. Myers, C. Vondrick, K. Murphy, and C. Schmid, *VideoBERT: A joint model for video and language representation learning*, Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2019, pp. 7464-7473.

[12] H. Wei, B. Ni, Y. Yan, H. Yu, X. Yang, and C. Yao, *Video summarization via semantic attended networks* Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32, no. 1. 2018.

[13] B. Mahasseni, M. Lam, and S. Todorovic, *Unsupervised Video Summarization with Adversarial LSTM Networks*, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 202-211.

[14] L. Zhou, J. Corso, and J. Xiong, *End-to-end Dense Video Captioning with Masked Transformer*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 2020.

[15] Z. Li, Y. Liu, H. Huang, and J. Yang, *Graph Convolutional Transformer for Video Summarization*, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020.

[16] S. Narayan, H. Wang, A. Alonso, M. Lapata, *Multimodal Transformer for Video Summarization*, Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2020, pp. 2704-2717.

[17] Zulko. *MoviePy Documentation*. GitHub. 2015. Available: https://github.com/Zulko/moviepy

[18] Anthony Z. *SpeechRecognition*. GitHub. 2017. Available: https://github.com/Uberi/speech_recognition

[19] M. Romero, *"BERT2BERT Shared Spanish Fine-Tuned Summarization Model*. HuggingFace. 2021. Available: https://huggingface.co/mrm8488/bert2bert_shared-spanish-finetuned-summarization

[20] X. Chu, Z. Tian, B. Zhang, X. Wang, and C. Shen, "Conditional positional encodings for vision transformers," arXiv preprint, arXiv:2102.10882, 2021.

[21] K. He, X. Zhang, S. Ren, and J. Sun, *Deep Residual Learning for Image Recognition*, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770-778.

[22] H. Yu, C. Chen, X. Du, Y. Li, A. Rashwan, L. Hou, P. Jin, F. Yang, F. Liu, J. Kim, and J. Li, *TensorFlow Model Garden*. 2020. Available: https://github.com/tensorflow/models

[23] T. Kipf and M. Welling, *Semi-Supervised Classification with Graph Convolutional Networks*, arXiv preprint arXiv:1609.02907, 2016.

[24] S. Chen, Y. Fang, B. Ni, X. Zhang, and J. Li, *Combining GCNs and Transformers for Video Understanding*, IEEE Transactions on Multimedia, 2021.

[25] Z. Wang, Y. Liu, Z. Yan, J. Yuan, and H. Yu, *Dynamic Graph Convolutional Networks with Attention for Video Summarization*, arXiv preprint arXiv:2012.00188, 2020.

[26] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” Advances in Neural Information Processing Systems, vol. 30, 2017. [Online]. Available: https://arxiv.org/abs/1706.03762

[27] W. L. Loh, “On Latin hypercube sampling,” The Annals of Statistics, vol. 24, no. 5, pp. 2058–2080, 1996.

[28] E. Tsalera, A. Papadakis, and M. Samarakou, “Comparison of pre-trained CNNs for audio classification using transfer learning,” Journal of Sensor and Actuator Networks, vol. 10, no. 4, p. 72, 2021.