AI-Driven Context-Aware Trimming and Segmentation of Educational Video Content for Focused Learning

by

Jonayed Kader Nabil
22101100
Tasnim Aziz Chowdhury
22101164
Sayed Ilham Azhar Harun
22101262
Rafi Ehtesham Mozumder
22301489
Nuzhat Tasnim
22101167

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science and Engineering

Department of Computer Science and Engineering
Brac University
January, 2026.

Declaration

It is hereby declared that

1. The thesis submitted is our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Jonayed Kader Nabil

22101100

Tasnim Aziz Chowdhury

22101164

Sayed Ilham Azhar Harun

22101262

Rafi Ehtesham Mozumder

22301489

Nuzhat Tasnim

22101167

Approval

The thesis titled "AI-Driven Context-Aware Trimming and Segmentation of Educational Video Content for Focused Learning" submitted by

1. Jonayed Kader Nabil (22101100)
2. Tasnim Aziz Chowdhury (22101164)
3. Sayed Ilham Azhar Harun (22101262)
4. Rafi Ehtesham Mozumder (22301489)
5. Nuzhat Tasnim (22101167)

of Fall, 2025 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science in 31 January, 2026.

Examining Committee:

Supervisor:
(Member)

Md. Golam Rabiul Alam, PhD

Professor
Department of Computer Science and Engineering
BRAC University

Co-Supervisor:
(Member)

Md Tanzim Reza

Senior Lecturer
Department of Computer Science and Engineering
BRAC University

Thesis Coordinator:
(Member)

Md. Golam Rabiul Alam, PhD

Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Sadia Hamid Kazi, PhD

Chairperson and Associate Professor
Department of Computer Science and Engineering
Brac University

Abstract

Recorded class lectures and tutorials are essential learning resources, but unedited raw videos generally contain significant non-instructional content, such as silence, administrative discussions, off-topic instructions, making effective navigation and revision difficult. This study presents a framework which is open-source, context-aware and can automatically detect and remove irrelevant or off-topic segments from long, unedited educational videos. Most of the existing related solutions rely heavily on large, closed-source models or fixed-length video segmentation with no context awareness, which are computationally expensive and prone to breaking semantic continuity. To address these limitations, we propose a modular, four-stage orchestration pipeline designed to run entirely on consumer-grade GPUs using small, open-weight models. The framework uses automatic speech recognition (Parakeet TDT 0.6B) to perform dynamic, sentence-level video segmentation, ensuring that each segment represents a complete semantic unit. A lightweight vision language model then generates structured textual descriptions for each segment using both visual frames and subtitles, augmented with rolling local context to preserve chronological coherence. Finally, a small language model (Qwen3-4B-instruct) is used to perform classification on the generated descriptions rather than on raw video input. The proposed orchestrator pipeline achieves **96.85%** accuracy, **89.04%** precision, **92.77%** recall, and **90.86%** F1 for irrelevant-segment detection on a human-annotated benchmark of 20 educational videos (total duration 09:13:47). This results in a significant improvement in content preservation over a one-shot Gemini-3 baseline, with precision increasing from 45.98% to 89.04% and accuracy from 81.63% to 96.85%. The entire system runs locally via quantized inference, offering a practical and privacy-preserving alternative to cloud-based solutions.

Keywords: Educational video analysis, Irrelevant content detection, Vision-language models, Automatic speech recognition, Context-aware video processing, Open-source machine learning

Acknowledgement

The authors thank Dr. Golam Rabiul Alam and Tanzim Reza, Department of Computer Science and Engineering, BRAC University, for their supervision and guidance throughout this research.

The authors also acknowledge the faculty members and the thesis coordination committee of the Department of Computer Science and Engineering, BRAC University, for their academic support and evaluation.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

The industrial and internet revolution have changed our way of living, consuming, behaving, learning and sharing knowledge. In the same way the AI revolution is also profoundly reshaping our life and environment. Especially learning anything became the easiest thing these days because of the large AI models trained on almost all the knowledge sources in the world. That's why these large models [19], [25], [27] have enormous knowledge which is based on pretty much every field and topic in the world. Large AI models have shown impressive knowledge and question-answering abilities but most of them are not open source even those with almost similar capabilities [33], [44]. They are usually computationally very expensive and not practical for using in standard hardware. In our case not feasible for local processing of long and unedited educational videos [17], [22].

In the modern era of digital education, recorded lectures and tutorial videos have become essential resources for students all over the world. These digital resources allow students to learn at their own pace unlike traditional classrooms. But the students face a big challenge in managing these resources effectively as the volume of video contents increases.

1.2 Rational of the Study or Motivation

Finding a moment where a specific concept is explained can be very time-consuming and frustrating for a student who is trying to revise that some specific concept. Most course contents or educational videos(webinar, live-streamed class, conference talks etc) are long and unorganised with some off-topic, irrelevant and unimportant contents. A typical video lecture can last over an hour and usually contains silence, repetitive explanations or irrelevant discussions which are not relevant to the core subject matter. This problem often leads to video fatigue where students waste valuable study time by scrubbing through footage in order to find relevant information[6].

The current methods for getting rid of irrelevant content within videos are inefficient and non-specialized. Most existing platforms those are capable of doing this depends on heavy closed source models which are very expensive and made for gen-

eral tasks [32], [45]. There are some works done in this area but those are usually for summarization or retrieval but most importantly they are not for educational or academical videos [20], [36], [37]. So there is a need for an intelligent system to exist that can bridge this gap and unlock the full potential of these educational resources without high computational costs.

1.3 Problem Statement

Long-form educational videos such as recorded lectures, tutorials, and webinars often contain significant portions of non-instructional content including silence, technical difficulties, administrative announcements, and off-topic discussions. Students seeking to review specific concepts must manually navigate through hours of footage, leading to inefficient study sessions and video fatigue.

Existing solutions for video content analysis rely predominantly on large, closed-source proprietary models [32], [45] that are computationally expensive and require uploading sensitive educational content to external cloud services. The existing solutions are not cheap and easily accessible to students and educators also not practical for institutions those have very high privacy concerns and tight budgets.

Moreover, existing video content processing and ingesting techniques usually uses fixed-length video segments which most of the time breaks semantic units by cutting in the mid-sentence, disrupting the coherence and flow of educational video contents[47]. Small-scale open-source multimodal models are fundamentally limited by their context window size and GPU memory requirements, making them incapable of processing long videos in a single pass [23], [24].

Additionally, these models alone lack the context awareness and reasoning capability to reliably distinguish between relevant instructional content and off-topic segments without specialized orchestration.

The core problem this research addresses is: **How can irrelevant segments be automatically detected and removed from long educational videos using open-source models on consumer-grade hardware, while preserving the semantic continuity and pedagogical value of the content?**

1.4 Objective

The objective of this research is to develop an open-source and context-aware framework for detecting and removing irrelevant segments automatically from long educational videos while also preserving meaningful instructional content. The system is intended to be efficient, scalable and usable on consumer grade GPUs. We focus on the following objectives in order to achieve this:

1. To design a dynamic, sentence-level video segmentation mechanism by using speech recognition and subtitles to split videos into meaningful segments instead of fixed-length cuts, so that explanations are not broken and the video makes sense.

2. To develop context-aware understanding by analysing both video frames and speech transcripts together while also keeping summaries of previous N segments so the system or specifically the small vision language model(VLM) understands each segment of the video flow through context.

3. To implement irrelevance detection by using small language model in order to classify each segment as relevant or irrelevant so the students get only useful content.

4. To scale to very long videos by processing video sequentially in manageable segments with local and global context which prevents long-context hallucinations by making sure that the model never exceeds its effective context window while also maintaining global awareness through main video topic and previous segment summaries.

5. To generate structured text descriptions and short summaries for each segment by creating a reusable index that enables searching, navigation and retrieval of video content in future extensions.

1.5 Methodology in Brief

The study provides a unique framework detecting irrelevant or off-topic segments from raw, unedited educational video. The system is built with small-scale, open-source language and vision language models which are deployable on consumer-grade GPUs that addresses the computational, economical and privacy constraints of cloud based solutions. The methodology employs a four-stage pipeline that integrates automatic speech recognition (ASR), vision language models (VLMS), and large language models (LLMS) in a cohesive workflow to overcome the limitations of fixed window approaches.

The existing video processing of fixed duration, scene change based segmentation and information lossy pixel budget problem is addressed by our proposed framework where dynamic, semantic segmentation strategy with manual frame sampling is used to content the coherence of educational videos. The workflow proceeds as follows with overview figure 1.1:

Stage 1: Intelligent Video Segmentation

For high accuracy sentence level segmentation and to avoid fragmenting semantic concepts, we are using the parakeet-TDT-0.6B ASR model[42]. Rather than providing arbitrary time intervals, this model has the ability of predicting precise sentence or segment timestamp with word duration, punctuation and capitalization detection which allows us to generate segment level timestamps based on natural linguistic boundaries. To capture non speech audio events for example silence, non human sounds, we further process this transcript using YouTube's free Closed Caption which helps the VLM to understand the environment better in stage 3.

Stage 2: Frame extraction and Multimodal Duplication

Using the timestamps from state 1, we are extracting frames based on 10% differentiation at 1fps rate on each timestamps, so that we can feed those frames as input

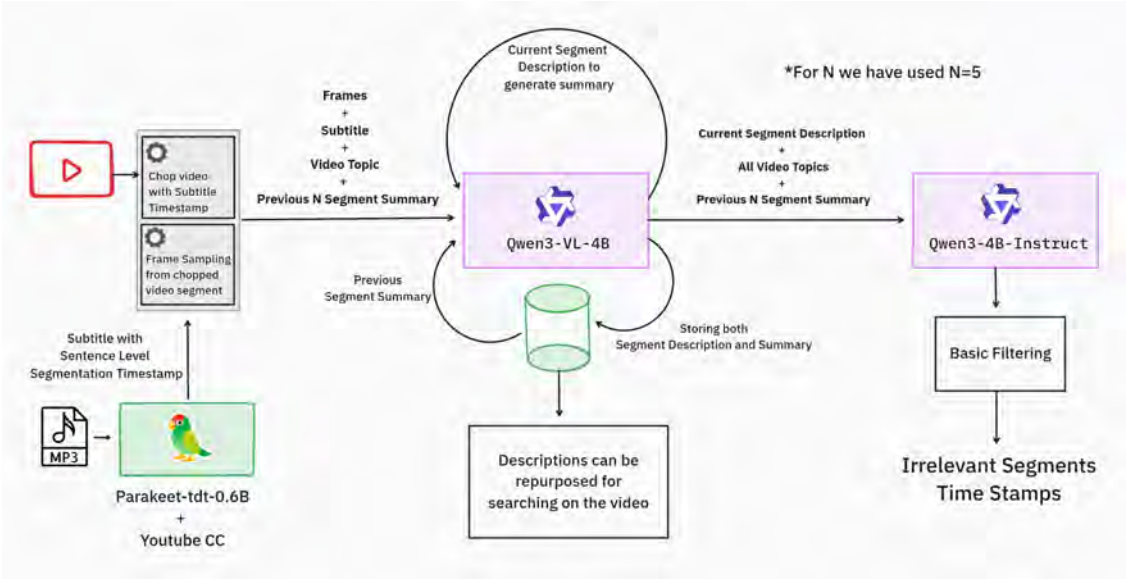

Figure 1.1: Top Level Overview of the Proposed Framework

to the VLM in stage 3. We are using perceptual hashing (pHash) for ensuring visual diversity without redundancy [3]. This process is done by calculating a unique "fingerprint" for each frame based on frequency domain features (Discrete Cosine Transform); frames with ham ming distance below a refined threshold are discarded as near duplicates. This clever approach reduces the token count for the downstream VLM processing while retaining all the unique visual information which works very well in educational settings.

Stage 3: Context Aware Segment Description

A small-scale VLM is used to generate detailed descriptions of each segment. In order to resolve the "Lost in the Middle" phenomenon [24], where model fails to retrieve information from the center of the long context, we are using local and global context in a small context window to solve this issue.

Each segment is processed with: Transcript from ASR model and the corresponding sampled visual frame. A global video topic. Summaries of previous N segments(N=5). As an output we get a description of each segment. This sliding context ensures that the model will maintain chronological continuity without crossing the limit of the effective context window which will eventually prevent hallucination, a common issue in long video analysis [35].

Stage 4: LLM as Final Classifier

A LLM is used for the final classification based on audio-visual environment heuristic. Speech is the primary element of an educational content; visual elements(slides, whiteboards) often persist on screen even during irrelevant digressions. Therefore when it comes to determining the relevance the LLM is instructed to prioritize the transcript or the audio cues first then visuals. The classifier defines a segment as "Relevant" only if the segment contains on-topic instructional content and filters out the silent, off-topic or noise with visual distractions [34].

This section presents a high level overview of the proposed framework. The detailed

design rationale, processing stages, and system implementation are discussed in chapter 4.

Data Collection and Analysis:

This framework is evaluated with a curated dataset of raw, diverse educational contents (Lectures, Tutorials, seminars) from publicly available YouTube videos. Which were human labeled irrelevant and relevant then post evaluated for further correction. Audio and visual data were processed locally, where sentence-level transcripts and corresponding video segments were generated and analyzed through a multi-stage pipeline. The outputs of each stage were examined through qualitative and quantitative accuracy of irrelevant and relevant segment detection and the preservation of instructional content. Performance is evaluated using standard metrics, including Precision, Recall, and F1-score, to assess the effectiveness of off-topic content detection.

1.6 Scopes and Challenges

Scopes

This research is bounded by the following criteria:

Education Domain Focus: This system is mainly focused on educational content where the content type is video lectures and information density is high. It mainly targets the common educational "noise" such as technical setup, Administrative announcement, off-topic student-teacher interaction, silent screen freeze and also non value part of the lectures; detects them as irrelevant.

Local, Privacy First Inference: All the components that are used to build the framework are open source models (Parakeet,QwenVL,Qwen, etc.) which are capable of running on single consumer grade GPUS (e.g. NVIDIA RTX 3000/4000 series). This scope excludes proprietary, API gated models like GPT-5, Gemini etc to ensure accessibility and reproducibility.

Long-Form Video Scalability: The solution can handle videos exceeding one hour in length. The dynamic segmentation and rolling context window of this framework bypasses the quadratic complexity scaling of traditional transformer attention mechanisms over long sequences.

English Language Constraints: The current ASR model that is used in our framework are only tuned for English Language. Cross lingual generalization is outside the scope of current framework but remains a potential avenue for future work. But the flexibility to choose models can somewhat resolve this language issue.

Constraints and Challenges:

The first challenge is the hallucination in small VLMs. Small parameter models (4B-8B) are prone to hallucination, commonly in case of "reasoning" about complex visuals scenes [35]. While large models can solve this problem, it will violate our hardware constraints. We address this by grounding the VLM strictly in the

provided transcript and the previous summaries of N segments, effectively replacing large visual interpretation with text.

A common challenge is the lost in the middle phenomenon. Regardless of the claimed context window of 32k or 128k tokens, it is quite commonly observed that performance degrades significantly when critical information is in the middle of the long input sequence [23]. The rolling summary approach of this framework entangles this by considering history into a cosine, adjacent prompt block, ensuring the VLM always attends to the most immediate and relevant context.

Another challenge and constraint is the dependency on ASR accuracy. The core Audio First segmentation and classification technique relies on the accuracy of the parakeet-TDT model. ASR word error rates (WER) increase when there is heavy background noise, strong accents, or overlapping speech (cocktail party problem) in the audio file, leading to poor subtitle. This erroneous audio transcript flow remains a significant challenge for the robust deployment. But it is mostly addressed by the YouTube cc integration with the parakeet model Here the YouTube cc(closed caption) is not open source but free to use.

The Subjectivity of "Irrelevance" is a big constraint and challenge to solve. The definition of the irrelevant topic might vary from person to person. A "personal anecdote" might seem unnecessary to some students but others might find it important to understand a topic. Capturing 100% true essence of a lecture with considering subjective definition of irrelevant is impractical but it is possible to extract key knowledge and specific explanation of a topic through removing all the known off-topic segments. That's why we try to detect only the most obvious irrelevant segments through some examples on the LLM instruction. So, even some irrelevant segments might be included in the relevant segments but we try to keep all the relevant and important segments as much as possible.
Another constraint and limitation is that our current framework skips silent teaching as it is instructed to prioritize the transcript and the audio cues. For example derivation of a formula. But the final form of any derivation or any important movement of the screen will be easily included in the relevant segments because after derivation it is obvious for the instructor to explain it.
These scope limitations and constraints are intentional design choices that directly reflect the project objectives and the practical limitations of deploying the system on consumer-grade hardware.

1.7 Thesis Organization

This thesis is structured into six chapters. The following is a brief overview of each subsequent chapter:

Chapter 2: Literature Review explores the foundational concepts of multimodal learning, Vision-Language Models (VLMs), and Automatic Speech Recognition (ASR). It provides a critical analysis of existing video summarization and filtering research, identifying the specific technological gaps in processing long-form educational content that this research aims to address.

Chapter 3: Requirements, Impacts and Constraints details the technical specifications and requirements of the proposed system. It also examines the societal, environmental, and ethical implications of the research, with a focus on privacy-first local processing and potential biases in small-scale models.

Chapter 4: Proposed Methodology presents the selected architecture in detail. It describes each stage of the four stage pipeline then explains how all the components work together. Comprising sentence level video segmentation, avoiding redundant frames, VLM based segment description and finally LLM based classification. It also explains type of collected data, data collection, cleaning and data stats. At the end of this section, we finish by explaining the technical implementation details.

Chapter 5: Result Analysis provides a quantitative and qualitative evaluation of the system using standard metrics like Precision, Recall, and F1-score. It compares the proposed orchestration approach against proprietary baselines (Gemini-3) and experimental fine-tuning attempts across a diverse benchmark of educational videos.

Chapter 6: Conclusion summarizes the key findings and highlights the contributions of the study. It also identifies the limitations of the current framework and proposes directions for future work, including multilingual expansion and user-personalized filtering.

Chapter 2

Literature Review

2.1 Preliminaries

In the current world, videos are one of the main content sources so their analysis requires multimodal learning. Multimodal learning is the system where multiple forms of information are processed together, such as texts, images, audio etc. Traditional AI systems only use a single mode of input, such as NLP uses texts and Computer Vision uses images. But we humans can combine multiple types of information at the same time to understand something. For example, listening to a lecture while reading a slide and also following what the lecturer is demonstrating. Multimodal systems are the closest thing that can replicate such human behavior/understanding. A huge challenge in such cases is to create a combined representation of all the modes of data. For example, a model must know how to connect the spoken words "supply and demand curve" with a graph that's being shown on screen, or even from multiple graphs on screen. The development of such embedded systems, where multiple types of information are closely coupled together, has been a significant step towards allowing models to perform difficult tasks such as finding images, videos based on a text description or vice versa [1], [2], [10].

Based on this principle, Vision-Language Models (VLMs) have emerged as the new groundbreaking technology. VLMs are multimodal models which can understand both visual and textual information. A typical VLM has two key components, a visual encoder and an LLM. The visual encoder of a multimodal model is often a Visual Transformer (ViT) which converts image or video data into a series of numerical representations, or tokens. These tokens are then fed into the LLM which has already been trained to make predictions and provide reasoning. By combining both of these parts, a VLM can perform tasks that are impossible for an uni-modal model for example, describing a video in natural language or identifying an object from a video. The process of combining visual and textual representation in the same latent space is crucial and is often achieved by extensive pre-training and post-training on a dataset with a huge number of image-text pairs. The application of these VLMs is a major area of research because generating natural language responses with image and text input creates an environment where an AI model can communicate with us in real-time, they can see us and talk with us.

The combination of the Automatic Speech Recognition(ASR) and VLM technologies has let the framework to understand educational videos like human do. ASR use

here for the "listening" part by translating the instructor's lecture into text even there are any interrupt by sounds or noice in the room. Models such as parakeet and Wishper [16] help in this process to produce fast and properly with accurate sentence breaks and the punctuation[42]. The VLM works for the "vision" part, which happens through understanding the frame, slides and the contents on the board. By integrating what it hears and what it sees, the system then can explain what is the lecture activities and then remove out the irrelevant parts. A Large Language Model is like a brain for our system, which is being designed here for performing the reasoning over the texts. LLM does not work on the raw video, in our system, instead of it, it analyzes the texts which are being structured by the ASR and the VLM. By using the reasoning ability, LLM can understand the meaning of an activity for example "roll call" and LLM marks it as an irreverent part. The LLM, based on Instruction-tuned allow rules for filtering decisions, enable proper and the context-aware reasoning [14].

2.2 Review of Existing Research

In the unedited or raw educational video, automatic filtering is quite challenging as these videos are not adequately handled by the mainstream video condensation technique. Most of the existing video filtering and summarization methods has designed for the visually dynamic videos like the series, movies, sports etc. On the other hand, the educational videos are mostly visually static, like the instructor showing the same slides for a long duration of time periods, and these videos usually contain condensed spoken explanations. Because of such reasons the usual video filtering and summarization technique is not appropriate or do not work well for the educational videos. To deal with these issues, our review discusses the different approaches from the existing research which are closely related to our problems solution and justifies the need or the importance of our proposed system.

In this writing we will explain where these existing systems fail and how our system offers a precise solution by using ASR + vision-language model based, multi-stage design which is more appropriate for the educational videos.

A recent paper is based on the visual-centric pipeline called LLMVS [38]. In LLMVS a video being segmented first through detecting the changes of visual scene. If the video scene is static like common in the lecture video then by default it breaks the video by a fixed time duration. Then the following segmentation a VLM creates a text based caption for each of the video clip. Lastly a LLM score the importance of each caption for final summarization. Though it is quite powerful in generic video summarization but without subtitle and environment understanding this visual-centric logic is basically unsuited for the educational videos noise of off-topic detection. Also in this method the LLM does not have any context about the video topic and context of chronological flow of the video.

A similar reviewed paper [7] which used GoogLeNet but with GAN and it chops videos using scene changes but still they do not have the scope to use any subtitle and video topic, which we solve by using subtitle and video topic.

A different kind of summarization paper [9] uses a full transcript of the video, use La-

tent Semantic Analysis(LSA) to summarize the text and then align the summarized text back to the video's timestamps. Our system's integrated multi-modal design is specifically architected to overcome this critical issue of this "decouple" method. Which is the loss of visual context. In a transcription-only method, the system has no knowledge of what's being shown on screen as the summarizer only has the audio transcription. It fails to distinguish between a segment where an instructor is explaining something complex from a slide saying nothing. Then it would never know what were on the board. But in our framework, we keep both visual data and language data tightly coupled throughout the process. For every segment, the VLM analyzes the video frames and subtitles altogether. This helps distinguish the two described scenarios above and then detect that the complex formula explanation segment is relevant and the story-telling is not. This is how our system sovles the transcription-only based systems problem.

Another reviewed work which also for video summarization is based on GoogLeNet and BERT for image-text representation [15]. This method basically partitions videos within short and fixed duration parts. Then it extracts characteristic from the visual and text streams to identify the important clips. Our method solves this partitions logic by using properly segmented subtitle which we get from the audio and use it as the priliminary driver for the segmentation. We are doing the segmentation based on the punctuation or sentence level segmentation not fix length segment. For the punctuation and sentence level segmentation we had used Parakeet-TDT-0.6B-v2/v3 [42], which is an automatic speech recognition model, it provides exact timestamp prediction and work on the punctuation restoration.

Most recent open source state of the art VLMs can do 1 shot solution with subtitle or Omni models can summarize or detect irrelevant and off-topic segments. But they have some issues when it comes to small sized models. For example, the VLM we are using Qwen3-VL 4B [31] does a dynamic pixel budget sampling where for longer videos in small context window it reduces frame resolution to fit on the context, which causes information loss. To run in local setup we can't just use very long context because of memory limitation. We address this by pre-sampling unique frames and using those frames instead of raw video.

Some key failure scenarios are evident the address our system.

The "Talking Head" Scenario: In many lecture videos, the instructor may have to stay in a single frame for a long time. For example, while explaining an important formula for a long time, and then all of a sudden, the instructor starts telling off-topic like personal stories. This will be a problem in LLMVS methods because it might treat the entire thing as one segment. Here, our system can fix this problem because our ASR-based segmentation does not focus on the visual things, instead it will listen to the speech of the instructor. So when we find a sentence change or an end of a sentence based on sentence level segmentation, like from academic to personal story sentence, a new segment will be created. This is how we can remove the irrelevant story from the important parts.

The Frame Missing Problem: As mentioned above as pure video input can be information lossy we can't use the default way in educational video settings as it

might miss crucial frames if the video is too long. So we solve this by pre-processing the segmented video by sampling unique frames at 1FPS. This has decrease the computational load at the same time keeping the main visual content untouched, ideal for standard local hardware.

The Isolation Problem: For the first 2 work they do not provide any previous or recent context of the video which causes loss of chronological of the video. Which might not be necessary for video summarization as they just need highlighted sections, but in our case we need to know the chronological flow so the LLM does not judge any segment in isolation. Because there might be multiple sequential segments which are irrelevant and the LLM can't judge them without the video flow and environmental model.

The "Mid-Sentence Cut" Scenario: When the GoogLeNet+BERT[15] architecture gets a long static video, it cuts the video after a fixed time like in every 60 seconds. This creates a problem, for example, if a sentence is not finished and it crosses 60 seconds, then the segment it creates does not carry any meaningful information, and the starting sentence is also meaningless. But in our case, we only cut on ASR-predicted sentence endpoints. As a result, there is no chance of creating any meaningless sentences, and it helps to keep the content clear and prevents errors caused by meaningless sentences. For improving the accuracy of the transcription, we had combined the Parakeet model (capture the boundaries of sentences) and the YouTube closed caption subtitle for non-human speech or audio tag.

In conclusion, by utilizing ASR-driven linguistic segmentation and coupling the visual and textual data, our system provides a better, more precise, reliable and context-aware solution for filtering educational videos than the existing state-of-the-art systems.

2.3 Summary of Key Findings

Few trends and limitations can be seen in the overall research field of video-language understanding. Firstly, it can be seen that the field has moved on from understanding video by tagging certain events that occur in the frames, to complex, generative reasoning. However, a major flaw of this approach is that models fail to process long-form content. Research shows that when the length of the content increases, the models suffer from "Visual Token Overload" and start to hallucinate. This tells us that applying such models on long-form contents is not efficient or reliable, more specifically when it comes to our long-form educational contents.

Another limitation is the way multimodal systems are integrated. It has been shown in studies that visual data alone isn't enough to explain what's being taught in an educational video, whereas the audio data carries most of the valuable information. Therefore, it is generally agreed upon that Automatic Speech Recognition (ASR) must be used to create high-quality transcription. These ASR transcriptions are more dynamic and respective of the speaker's flow of speech, which result in better information retrieval.

Research shows, it is better to describe and then reason instead of doing both at the same time. A high parameter cloud-based model might work well but it has privacy issues and it is quite expensive. For this reason recent works choosing small models and techniques such as the frame sampling, model quantization. So that it can be run on the normal GPUs.

The implication of the findings of our thesis reflects that educational video's filtering should use modular architecture and memory loop to avoid the context collapse. Analyzing the structured text rather than working on the raw pixels will be the best way to handle the lengthy educational videos. This creates a safe-privacy, local and multi-stage pipeline which can reduce the gap between the logic and the perception. Overall, the outcome reflects that our proposed work is both needed and possible for overcoming the identified limitations of the current lengthy videos.

Chapter 3

Requirements, Impacts and Constraints

3.1 Final Specifications and Requirements

Our main specifications and requirements are as follows:

- The system must be able to identify and trim irrelevant or off-topic segments from educational videos.
- All the models used must be free, open-source and open-weight.
- Models must be small enough to run locally on a consumer grade GPU.
- Can scale to process very long videos.
- Flexible design so that user can use state-of-the-art small open models.
- User can use the whole video information for further question-answering, summarization and retrieval.

3.2 Societal Impact

As we have mentioned in previous sections, our project aims to democratize access to advanced video analysis tools for students by leveraging open-source models that can run on consumer-grade hardware. So that students do not need to rely on expensive closed-source solutions and models. Which aren't even specifically designed for educational video analysis purposes. By providing an accessible and affordable solution, we hope to empower students from diverse backgrounds to engage with video content more effectively, enhancing their learning experiences and academic performance.

Also, as our framework is open-source, anyone can use it for their video analysis and context stitching tasks. By making our work open-source, we let anyone who is willing to work further on this project have access to everything, and also we contribute to the overall community of video understanding analysis. Thus, we eventually contribute to the advancement of technology, knowledge and education.

3.3 Environmental Impact

The main focus of our thesis is to make the framework run on consumer grade GPUs which are not very high end in terms of resources. By doing so, we are technically reducing the use of high resources, which is impacting negatively on the environment in the current world. For example, high usage of electricity or emission. With the use of smaller models, we are reducing the carbon footprint of our project, along with the ones we are replacing. Moreover, by running our framework on the consumer-end, we are reducing the use of cloud-based services, which tend to use a tremendous amount of resources.

3.4 Ethical Issues

By running our framework locally, we are respecting the need for privacy for students who will be using our framework. Maintaining privacy is a huge concern in this era of digital information-exchange and our framework helps protect just what our users don't want to share. Moreover, the open-source models which we are using for our framework – Parakeet TDT 0.6B, Qwen3-VL and Qwen3-4B-Instruct-2507 [31], [41], [42] have all been developed following ethical AI practices. By using these models, our framework ensures that there is complete transparency and accountability of the data that is being used in the framework. Additionally, our framework detects segments that are irrelevant in a video. So it can help remove biases and any discrepancies in video analysis. This ensures that the outcome is always more accurate and reliable. The Parakeet TDT model was trained on English language and most of the data needed for it to be trained was Western American English. So it may be biased towards western videos and may make mistakes detecting non-western accents or dialects even though it is one of the small state-of-the-art models with only 0.6B parameters.
Also even though Qwen3-VL and Qwen3 4B Instruct models are trained on multi-modal and multi-lingual datasets and state-of-the-art in this size, they may still hallucinate and have biases towards certain languages or cultures depending on the training data distribution. That's why we have used 2 shot or layer approach and force the Models to focus on irrelevant or off-topic segments only so we can avoid hallucinations and trimming relevant content.

3.5 Standards

The project follows established standard practices in software engineering and machine learning experimentation. Open-source frameworks such as llama.cpp, Unsloth, Hugging Face Transformers etc were used during development and experimentation, These tools and frameworks adhere to widely accepted community standards for reproducibility, documentation, and responsible usage. The models employed in this work are open-source and licensed for personal and research use.

System evaluation was conducted through human-based benchmarking, where classification outputs were reviewed and validated manually. Our approach the standard evaluation practices in AI/ML research where AI system helps human in educational environment. We also ensure that performance evaluation reflect practical usefulness rather than relying solely on best case scenarios. Although we have not followed any formal ISO or IEEE standard in our work, the overall development and evaluation process conforms to recognized academic research norms. We have tried to maintain as much as possible transparency in our methodology, ethical data usage and reproducible experimentation.

3.6 Project Management Plan

To deliver a proposed system that provides only the relevant segments from an educational video, we need to follow an iterative and modular process. This approach reduces integration risk by validating each module individually and independently in each iteration. The project can be considered complete when it can deliver a list of segment timestamps or a trimmed-cut video of the original video which only consists of the relevant segments. It should also be able to store aligned transcripts, segment descriptions and so on to support further integrations of new features in the future.

Final Specification and requirements were created based on the deliverables of functional and non-functional requirements. These were making sure the final output is a list of segments which are relevant in the video, making sure the framework runs locally and using models that are open-source. We also have to make sure that the context doesn't grow larger than the local computational limit. For which sentence-level segmentation, deduplicated frame sampling and rolling context summaries are used.

In development, we followed iterative development, and each iteration was driven by the test results of the previous iteration. Early iterations showed that the model was not highly performative when it came to long videos due to the limitation of contextual constraint on limited hardware resources. This invokes us to move on to a two-shot approach, where the VLM first generated description of the segments and then the descriptions were used to find irrelevance in them. This strategy helped us reduce context requirements and also increase stability while keeping the inference within the limits of local hardware execution.

In terms of resource planning, we had to account for team members, hardware and software resources, and the dataset. Team members had been distributed with the responsibilities of finding suitable videos to run on the framework, then curating the dataset, labeling the segments, reviewing generated descriptions and segment timelines, benchmarking and evaluation, and then finally, report writing. We also had to run our inferences on consumer grade GPUs, but for experimental finetuning, we had to take help from the Research Lab of BRAC University.

Finally, after all the experiments and inferences, we summed up our work and documented everything by writing drafts of our research experience. We recorded major changes which included the need to swap models for better results, or deciding rules for segmentations, adjusting prompts etc. When delays had occurred, we resched-

uled activities and kept our core deliverables as the main priority and delayed the optional enhancements such as searching and dataset enlargement as future work. This helped us maintain the project expectations within the given timeframe.

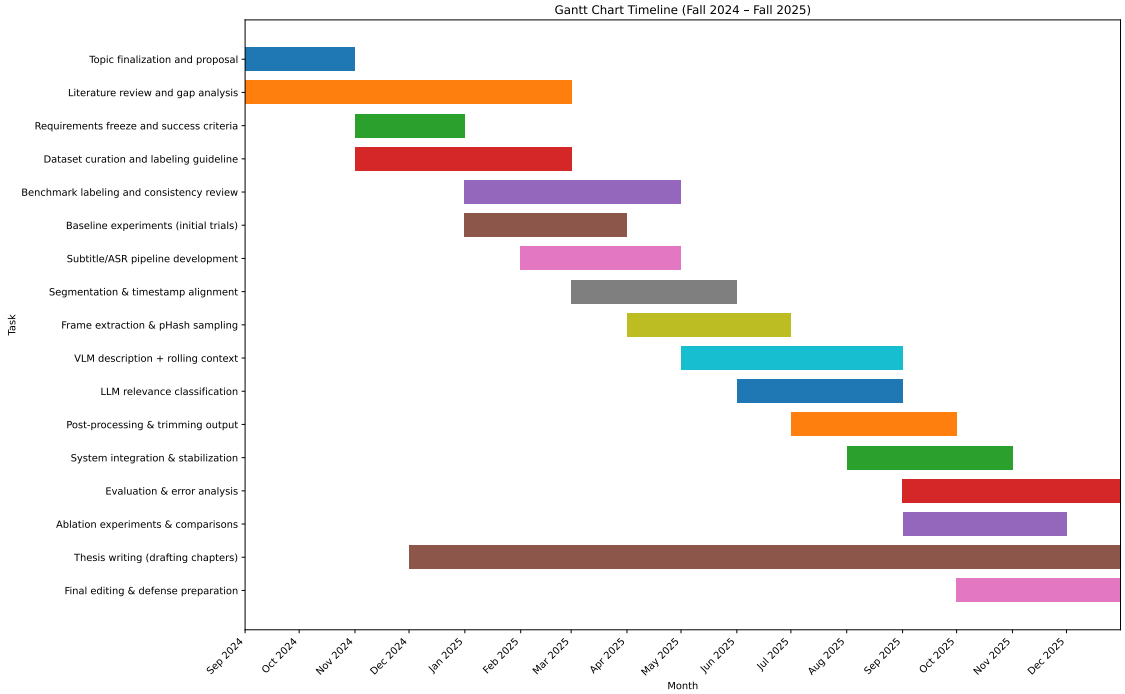

Figure 3.1: Gantt chart timeline for the thesis (Fall 2024 to Fall 2025).

Table 3.1: Budget estimate (BDT).

Budget Item	**Description**	**Estimated Cost (BDT)**
Software licenses	Open-source and open-weight tools	0
Compute	Local GPU inference and experimentation (in-kind)	0
External storage	External HDD/SSD for videos, frames, backups	3,500
Printing and binding	Final submission printing/binding	1,200
Miscellaneous	Internet/electricity contribution and stationery	1,000
Total		**5,700**

3.7 Risk Management

This section identifies, assesses, and manages risks that may affect the project's technical quality, schedule, or successful completion. Risk management is necessary because the pipeline consists of dependent stages where early-stage errors, such as transcript alignment issues or segmentation mistakes, can propagate into description generation and relevance classification. Risks were assessed qualitatively using Low, Medium, and High for both likelihood and impact. Mitigation actions reduce the probability or severity of a risk, while contingency actions describe recovery steps if the risk occurs.

Risks were identified from baseline trials, integration testing, dataset preparation, and practical constraints of local inference. Technical risks include hallucination-driven trimming errors, long-context instability, and GPU VRAM limits. Data risks include ASR inaccuracies and ambiguity in relevance labeling. Integration risks include timestamp drift and schema mismatches. Schedule risks include delays caused by repeated experimentation and tuning. Risk monitoring was performed during weekly reviews and after major configuration changes, and any high-severity recurring issue triggered immediate mitigation or contingency action along with an update to the project plan.

Table 3.2: Risk register (likelihood, impact, mitigation, and contingency).

Risk	**Likelihood**	**Impact**	**Mitigation (Prevention)**	**Contingency (Recovery)**
Model hallucination trims relevant lecture content	Medium	High	Use two-stage reasoning (describe then classify), conservative prompting, and post-processing rules to avoid aggressive removals	Add a verification pass for borderline segments and restrict trimming to clearly irrelevant blocks for final deliverables
Long-context instability causes inconsistent decisions	High	High	Use sentence-level segmentation and rolling summaries; avoid feeding long raw content in one call	Reduce carried context, shorten summary windows, and increase segmentation granularity
ASR errors reduce segmentation quality and downstream accuracy	Medium	High	Apply transcript sanity checks, caption merging where available, and silence tagging	Switch ASR variant, apply basic audio cleanup, and exclude severely degraded audio cases with justification
Mid-sentence cuts reduce coherence and harm trimming quality	Medium	Medium	Clamp boundaries to punctuation/sentence rules and enforce minimum segment length	Merge adjacent segments and re-align boundaries to the nearest valid punctuation/timestamp
GPU VRAM constraints prevent running chosen configurations locally	High	High	Use quantized models, strict frame budgets, and deduplication sampling	Switch to smaller models, reduce resolution/frame rate, and run rare edge cases on higher-VRAM hardware if available
Frame sampling misses critical slide/board information	Medium	Medium	Use diversity-aware sampling and ensure a minimum number of representative frames per segment	Increase frame budget for visually dense segments and rely more heavily on transcript evidence in those cases
Benchmark labeling inconsistency due to subjective relevance	Medium	Medium	Define labeling rules and perform consistency review	Remove ambiguous segments from quantitative scoring and report them separately as limitations

3.8 Economic Analysis

The economic cost of this project is relatively low due to the exclusive use of open-source models, frameworks, and tools. Libraries such as llama.cpp, Unsloth, HF-Transformers, etc were used without licensing fees. The employed models are pre-trained and freely available for personal and research use. This significantly reduces software and training costs compared to proprietary solutions.

The primary cost associated with the project for experimentation is inference and fine-tuning computation. By selecting lightweight models and performing local inference, the system avoids recurring expenses related to cloud computation and data transfer.
Additionally, for experimentation even though we did LoRA[12] fine-tuning which is extremely efficient and done locally on a single GPU cut our experiment cost by a lot. As the model were so generalizable, the system did not need for fine-tuning or retraining at the end. Which makes our solution more flexible and constrained to any specific tuned model.

Overall, the proposed solution offers a cost-effective approach for educational video analysis, making it suitable for individual students with resource-constrained environments.

3.8.1 Cost and Benefit Analysis

In this section, the overall costs and benefits of our proposed framework have been analyzed. The analysis is basically based on development, operational cost and for the sake of long-term benefits for the educators, students. As our proposed system can being run on locally using open-source models so the financial cost are minimal, whereas the output is high practical value. In our system, the main cost is based on the hardware requirements, such as the system is designed to use on a consumer-grade GPU for example NVDIA RTX 3000 series with approximately 12GB of VRAM. Such hardware is very common in many universities' labs, research labs and also for personal use as well. No cloud infrastructure or specialized setuo needed for this system. So, it is quite visible that the initial investment for the system is quite low compared to the cloud-based models. One of the vital cost factor is software. All the models and tools being used in this system such as the Parakeet TDT, Qwen3-VL, Qwen3-4B-Instruct are free to use as they are open-source. There are no subscription charges or API usages for them. This reduces recurring costs which are mostly needed for the commercial AI services.

The system has development and maintenance costs. Efforts and time are needed to combine different models and tools for example ASR model, frame extraction, VLM based description, LLM based decision on one pipeline. When the system is set up then the required maintenance cost is minimal. For the modular design, individual components can be updated without redesigning the full system again. For this the long-term maintenance effort and the cost being reduced.

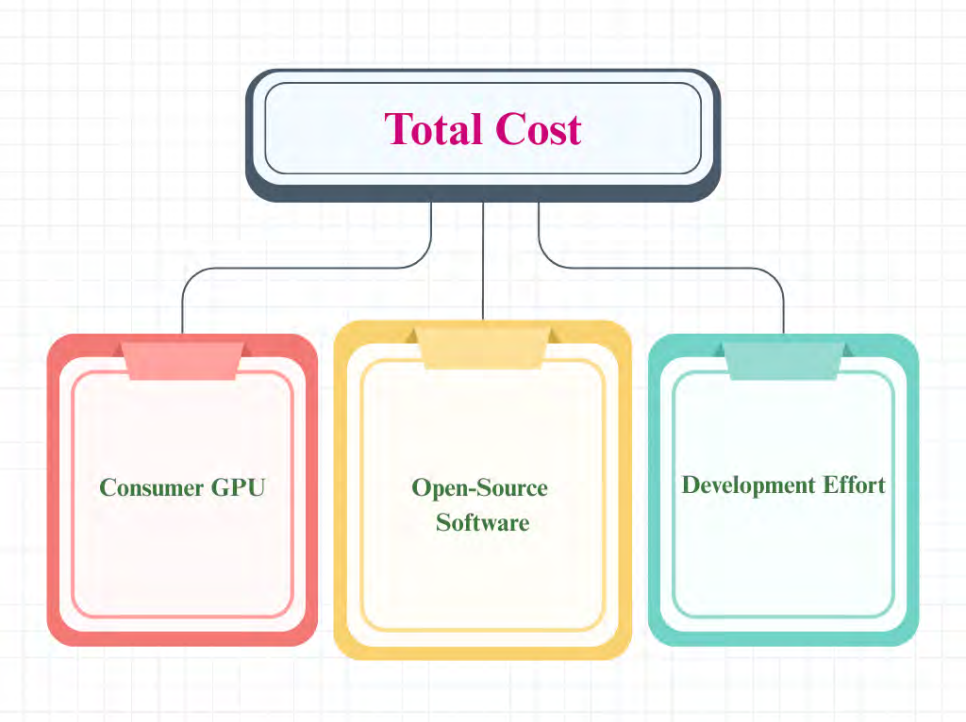

Figure 3.2: Total Cost breakdown for the proposed framework

The proposed system give significant value to the educator, students by reducing video fatigue and time. Instead of manually searching in the long duration lecture videos, now students can focus on the relevant and instructional parts. This will improve learning efficiency that will help the students revise the topics more efficiently. For the educator our system will open a new way to enhance the quality of the recorded classes. No manual editing will be needed to make educational content more structured. Institution can be benefited from our system as it the system is privacy-safe and local development based. Here confidential or sensitive educational data not needed to be sent for to external servers, this maintains the data protection policies.

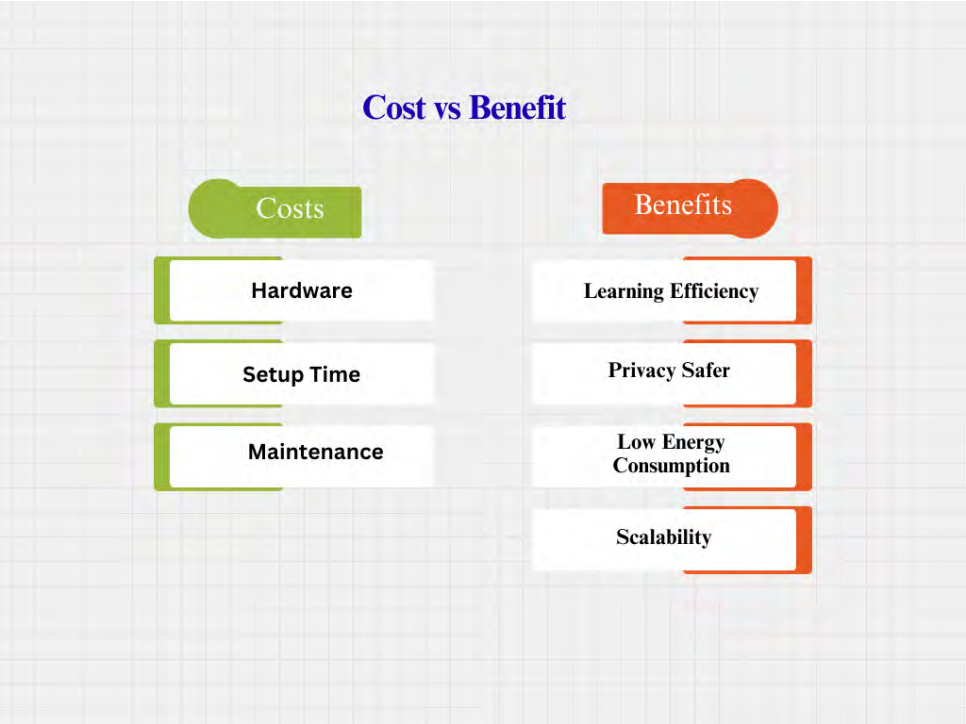

Figure 3.3: Comparison between operational costs and educational benefits

Overall the benefits of our proposed system is clearly costs effective. By integrating low hardware requirements, no subscription fee, zero licensing fees, quality educational value, oursystem provides low-costs, sustainable and scalable solution for the educational video analysis.

Chapter 4

Proposed Methodology

4.1 Methodology Overview

Our main objective is to get rid of irrelevant or off-topic segments from videos using small open-source models that can run on consumer-grade GPUs. To achieve this, we had to fight with 2 main constraints. First one is the small size of the models we are using. We have analyzed benchmarks like MMLU-Pro, TruthfulQA, GPQA etc [13], [26], [28] and experimented that small models tend to hallucinate and make mistakes more often than large models. Also from the physics of language models paper [29], we know that language models store around 2bits of information per parameter also other scaling law papers like [8], [11] support why large scale models are better. So smaller models have less capacity to store information and knowledge. That's why we opted to use state-of-the-art small models like Qwen3-VL, Qwen3-4B-Instruct-2507 and Parakeet TDT 0.6B [31], [41], [42], which are some of the best open-source models in their size category. Additionally, we used 2 shot or 2 layers of models to improve the performance of the small models.

The second and a big issue is processing long videos. We know at the end of the day LLMs are big nerual networks and they have a context window limit. Even though models like Qwen3-VL, Qwen3-4B-Instruct-2507 have a large context window support but it hallucinates or accuracy drops when we feed them long inputs experimented in paper like "Lost in the Middle","What's the Real Context Size of Your Long-Context Language Models?" [23], [24]. Also, inspite of having large context window support we can't utilize the full context window due to GPU VRAM limitation. So we can't feed a very long video to the model at once, even large models like GPT-5, Claude Opus 4.5 has only 128k-400k context window and large videos can't be processed at once. Only Gemini 3.0 models has 1M context window but it's not open source and the token price is very high for specially for long videos it takes a lots of tokens and it is very expensive. To solve this problem we need to keep the context window size as small as possible along with providing enough context to the model. So we need to break the video into smaller chunks and process them one by one and that's what we did.

But chunking introduce 2 more issues:
First one is how should we chunk the video? We can't just randomly chunk them, then we might just cut the middle of a sentence or a word. The second one is how the model would understand the context of the video with only a small chunk of it.

To solve the first issue we needed sentence level segmentation of the video which has to be meaningful but not too long segment(group of sentences) also not shorter than a sentence. Here comes Parakeet TDT 0.6B ASR model. It is a state-of-the-art small ASR model that can generate sentence level segmented subtitle of audio with auto punctuation and capitalization support. But it can't detect non-human sounds, that's why we used youtube's closed caption subtitle for non-human sounds and merged them with ASR subtitle. We also consider the gap or non-speech segments as silence and inject "silent for Ns" caption between the segments to make it a near perfect well segmented subtitle with proper timestamps. Now we use this subtitle to chunk the video into smaller meaningful chunks which then fed into the VLM.

But there is a small efficiency issue during feeding the video chunks into VLM. As the video chunking is now dynamic the chunk can be of different length and the Qwen3VL sample frames and decide tokens/frames dynamically based on the chunk length and video resolution. In our case we can't miss any important information/frame from video as we are working with educational video. Also we can't just feed the arbitrary length of video which need a lots of GPU memory and in our case we can't afford that along with we already know the problem of very large context problem in these small models. So, to balance the efficiency and accuracy we do our own sampling of frames from the video, we sample unique frames which are less than 90% of the previous frame at 1 FPS. That's how we are not missing any important information from the video while keeping the context window size small.

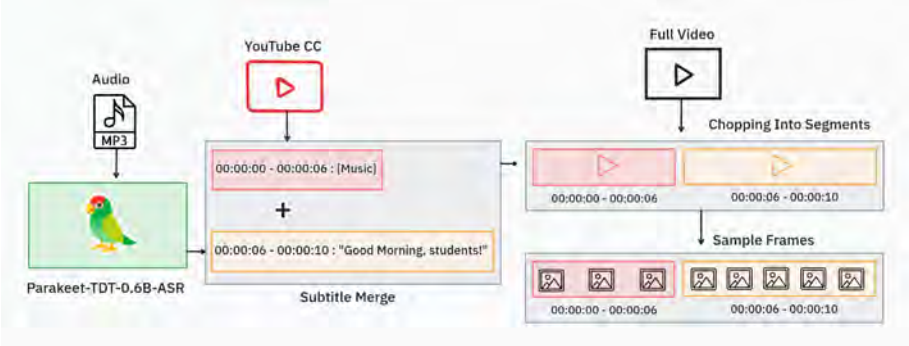

Figure 4.1: Dynamic video chunking and unique frame sampling process

Now, to solve the second issue we needed to provide enough context to the model to understand the context of the video. We are solving this by context engineering the local and global context and providing them to the model along with the video chunk frame and subtitle. For global context we provide the whole subtitle of the video to the LLM then it provides us the overall topic of the whole video. We do it twice first time for VLM input which is the main topic(shorter version) and second time it is every single topics of the whole video. This way the model gets to know what the video is about and what are the different topics in the video. For local context we provide textual description summary of previous N segments to during inferencing of current segment/chunk textual description. By providing the previous segments description summary and global context along with the video chunk frames and subtitle we stitch the context such way that it requires less context window size but provides enough context to the VLM to understand the video flow. Basically we are

trying to simulate the human watch a video. When a human watch a video he/she already has the overall idea of the video topic and also remembers the previous few sentences/segments.
So during getting the textual description of a segment we provide the following inputs to the VLM:

- Video chunk frames (sampled frames from the chunk)
- Video chunk subtitle (subtitle of the chunk)
- Global context 1: Overall topic of the video
- Description summary of previous N segments (local context)

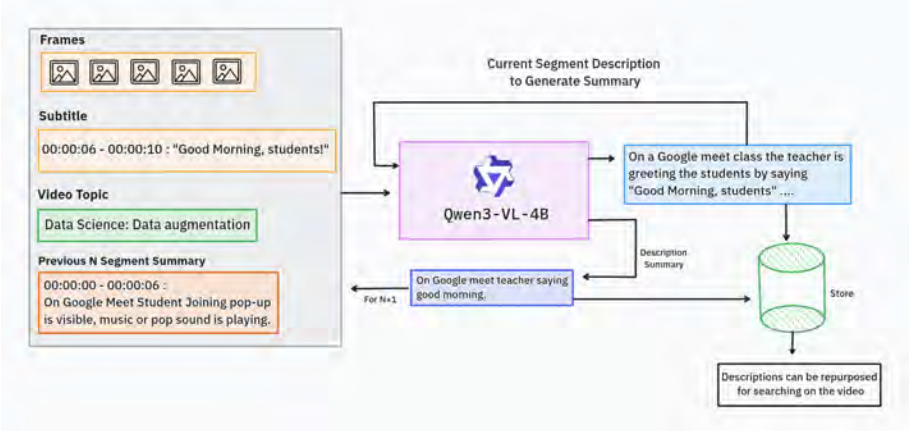

Figure 4.2: Input and output of the VLM stage for segment description

How we get the description summary of previous N segments? We do it in 2 steps. First we get the textual description of a current segment then we send this description to the VLM to summarize the textual descriptions of that current segment to get a concise summary. This summary is then used as local context for the next segment/chunk to get its textual description. We don't use the whole previous N segments description as it will take large context window and also it will be redundant information for the model. So we use the summary of previous N segments description to keep the context window small.

In this VLM stage we don't classify the segment as relevant or irrelevant, we just get the textual description(also summary) of the segments and store them. Because classifying relevant or irrelevant segment very hard for small models because of its reasoning capabilities for multistage tasks. How do we know that or why? Because it was our initial approach to classify the segments in this stage but the accuracy was very low.

Now to classify the segments as irrelevant or relevant we use another small LLM Qwen3-4B-Instruct-2507 [41]. In this stage we offload the classifying task to this small LLM which is specifically instructed for classification tasks. We provide the following inputs to this LLM during classification:

- Textual description of the segment with subtitle (from previous VLM stage)

- Every single topics of the video (global context)
- Previous N segments description summary (local context)

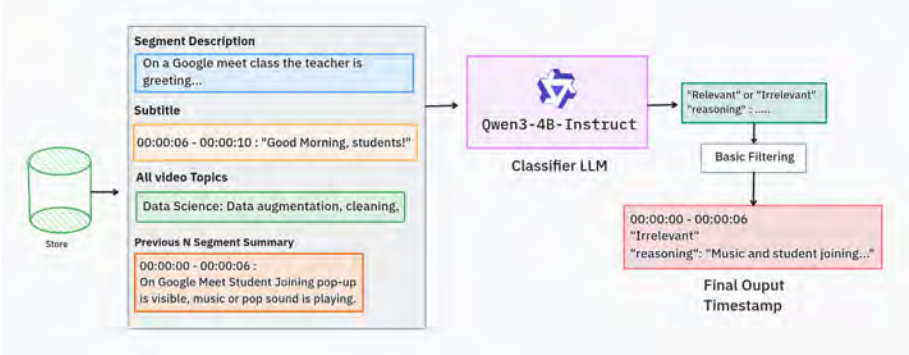

Figure 4.3: Input and output of the LLM stage for segment classification

For N we experimented with different values like 2,3,4,5 and found 5 is the optimal value for our use case.
After getting the classified segments we do some basic post-processing and filtering to merge the adjacent irrelevant segments together. Also remove very small, isolated irrelevant and relevant segments.
Finally, we get the final irrelevant segments timestamps which can be used to trim the original video to get the final output video.
As we have the textual description of the whole video, we can use it to generate summary, do question answering and even retrieving or searching specific segments from the video using vector embedding based searching which we also experimentally implemented to explore the possibility.
Now if we summarize the whole design process in steps:

1. Get sentence level segmented subtitle of the video using Parakeet TDT 0.6B ASR model and youtube closed caption.
2. Chunk the video into smaller meaningful segments using the subtitle timestamps.
3. For each segment get the textual description using Qwen3-VL with local and global context.
4. Summarize the textual description of each segment to get concise summary for local context.
5. Classify each segment as relevant or irrelevant using Qwen3-4B-Instruct-2507 with local and global context.
6. Post-process and filter the classified segments to get final irrelevant segments timestamps.
7. (Optional) Use the textual description of the whole video for summary, Q&A and retrieval.

Here is a high-level diagram of the whole pipeline:

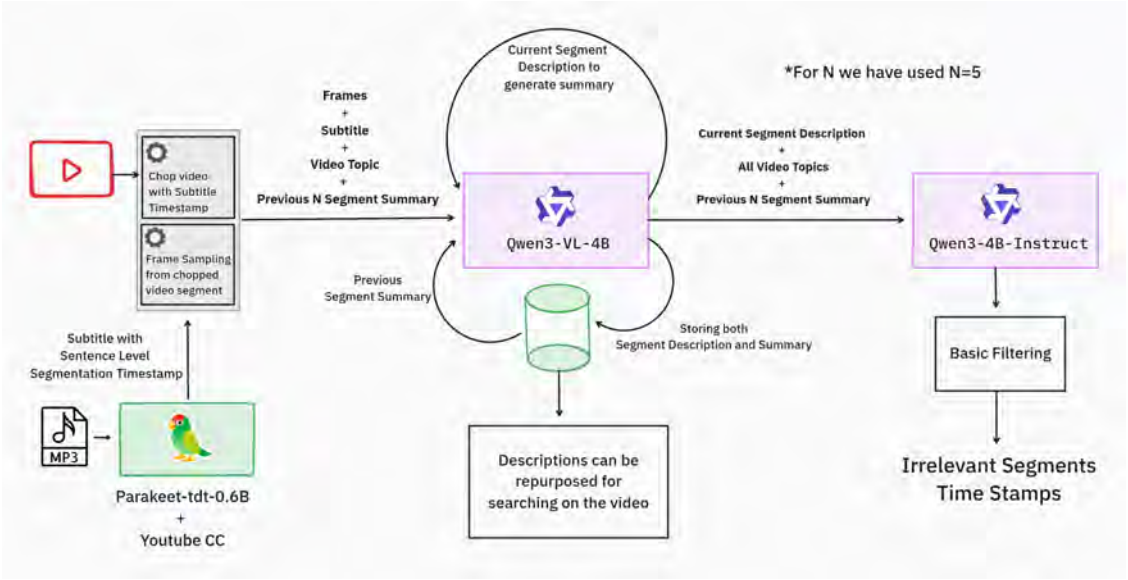

Figure 4.4: Our whole orchestration framework

4.2 Preliminary Design

As discussed in the previous section the first thing we have tried is to directly classify the segments as relevant or irrelevant using only VLM. On that approach we tried using overlapping segments to provide the contexts but as we were segmenting the video in a fixed length according to context window size. But the overall output were not acceptable, even we have tried different small VLMs like InternVL-2.5, Qwen2.5-VL, SmolVLM-2 etc [21], [30], [39].

Figure 4.5: Initial approach using single-stage VLM classification

Then we noticed the VLMs we have used are not good at understanding the environment. So we decided to improve the VLM understanding capabilities by training it with more educational video data. So we have collected more educational video data and full and LoRA fine-tuned the VLM with that data along with irrelevant segment examples. But still the performance was not satisfactory.
Then we have analyzed the problem and found that classifying relevant or irrelevant segment along with understanding the environment is a complex task for small VLMs. Also the overlaping segments approach was not working well as it was cutting the middle of sentences and important information were missing.
So we have decided to split the task into 2 stages, first stage is to get the textual description of each segment using VLM with proper local and global context engineering. Then in the second stage we use a small LLM to classify the segments as

relevant or irrelevant using the textual description from previous stage along with local and global context.

Also we have decided to use dynamic chunking of the video using sentence level segmented subtitle and instead of overlapping segments we provide local context using previous 1 segments description summary.
In the begining we the LLM were using Gemma 3, Llama 2 models [18], [43] but the performance were not good enough. So we collected textual description data and fine-tuned those LLMs but those models still couldn't understand the environment well enough to classify the segments accurately.
Later we switched to Qwen3-4B-Instruct-2507[41] and it worked really well and exceeded our expectations. As we already collected the text data for classification task we did some LoRA fine-tuning adding some more diverse semi-synthetic data for experimentation.
But the Qwen3-4B-Instruct-2507 model was already very generizable because of its very good post-training methods, that even its prediction on our train data were very well we can see the how the loss curves were falling so quickly. Here is one of the curves of different fine-tuning experiments we did with Qwen3-4B-Instruct-2507:

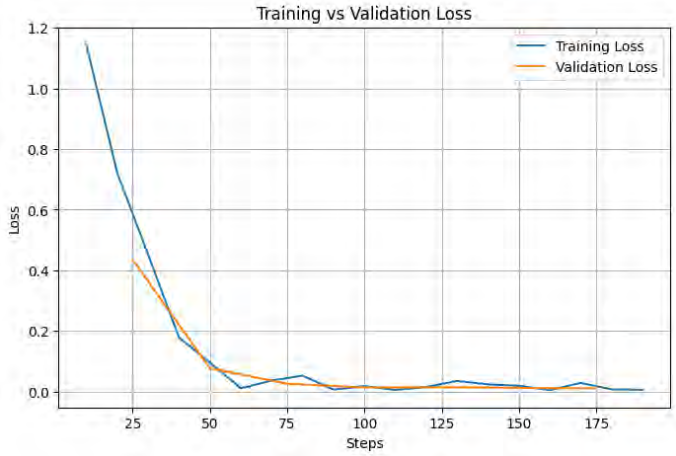

Figure 4.6: Training loss curve from fine-tuning experiments with Qwen3-4B-Instruct-2507

So finally we have arrived at our final design process as described in the previous section and decided not to fine tune it further as the fine tuning plan, data were for the previous models but this model there is no need for fine-tuning. Here are the data we have used for fine-tuning the previous models:

4.3 Data Collection

Two distinct datasets were collected to support different phases of the research: a benchmark dataset for system evaluation and an experimental dataset for model fine-tuning experiments.

4.3.1 Sources of Data and Methods of Collection

Benchmark Dataset

The benchmark dataset comprises 20 publicly available educational videos sourced from YouTube, spanning diverse type of educational videos including recorded "zoom" or "google meet" class, recorded live-streamed online class with students in chat and recorded seminar. Videos were selected based on some criteria. Such as "Educational Nature" which means content must be instructional with clear teaching objectives. Also, content must be Variable Length so he videos range from 3 minutes to over 1 hour to test scalability. Then most importantly, videos must contain identifiable off-topic segments such as technical difficulties, administrative announcements, or tangential discussions. As we are using ASR so, we need to ensure sufficient or minimal audio clarity for accurate speech recognition.

After collecting the videos, we manually annotated each video by human reviewers who labeled segment timestamps as either "Relevant" (on-topic instructional content) or "Irrelevant" (off-topic, silence, or non-instructional content). A secondary review process was conducted to resolve ambiguous cases and ensure labeling consistency and accuracy. During the 2nd review process for very ambiguous irrelevant case we discussed with the labelers and then decided to keep it or discard it as we don't want to remove any slightly relevant content.

The Benchmark link: EduCutBench dataset (Hugging Face)
https://huggingface.co/datasets/znyd/EduCutBench

Experimental Dataset

The experimental dataset consists of textual descriptions of video segments extracted from educational content, used for fine-tuning classification models. Segments were described based on their visual and audio characteristics, including:

- Scene description (presenter speaking, slides displayed, whiteboard content)
- Audio transcript summary

The dataset link: Irr-Rel-mix-new dataset (Hugging Face)
https://huggingface.co/datasets/znyd/Irr-Rel-mix-new

4.3.2 Data Cleaning

For the Benchmark Dataset: Some cleaning procedures were applied to the benchmark dataset. Firstly, we removed very short isolated irrelevant segments which were just unnoticeable noise. Also, very small relevant segment between with adjacent to large irrelevant segment were merged with the irrelevant segments.

Then segment boundaries were adjusted to align with natural speech pauses to ensure clean cut points. Segments with unclear relevance (e.g., contextual anecdotes) underwent secondary review with multiple annotators. If labeled irrelevant segments group those are very close showed small zig-zag pattern then they were re-evaluated.

For the experimental classification dataset: Incorrectly written descriptions were identified re-checked through cross-validation and corrected. Duplicate or near-duplicate descriptions were removed to avoid data leakage. Grammatical inconsistencies in textual descriptions were re-written and standardized.

4.3.3 Data Transformation

Benchmark Data Transformation

The benchmark ground truth was transformed into a structured format. Video segments were encoded as binary labels as "Irrelevant" or 1 (positive class) and "Relevant" or 0 (negative class).

Normalized all timestamps and converted to seconds along with hours:minutes:seconds format from start to end of the video for consistent processing. Then the final ground truth was stored as JSON file with video ID(YouTube ID) and all the irrelevant segments timestamps along with segment index.

Experimental Data Transformation

The experimental dataset underwent the following transformations:
Segment descriptions were formatted with consistent structure (visual description, audio transcript, context). Binary classification labels applied (Relevant/Irrelevant). Each sample stored with metadata including source video, scenario and topic category.

4.3.4 Data Integration

Experimental Dataset Augmentation

To increase dataset diversity and challenge model generalization, semi-synthetic data was integrated with the original experimental dataset: Additional irrelevant scenario descriptions were generated by adding new scenarios, video topics which were not present in the original dataset and it covers a lot of edge cases. For the new topic we created different kind of scenario description through another large LLM[40]. Then we paraphrased the existing scenario description ambiguously to create more hard to detect irrelevant scenario descriptions.

We also applied hard negative mining where borderline cases (segments that are contextually relevant but appear off-topic in isolation) were specifically collected to challenge the classifier. We did source balancing through collecting data from multiple video sources to prevent model bias toward specific presentation styles.
The integration increased dataset size approximately to 7K data point or segment scenario.

4.3.5 Data Reduction

Given the nature of video data, several reduction techniques were applied to manage computational requirements:

Firstly, manual frame sampling rather than default way of processing raw video frames, unique frames were sampled at 1 FPS with perceptual hash-based duplication (frames with <10% visual difference removed). It reduced the number of input frames also very important for accuracy in education videos.

VLM output of long segment descriptions were summarized to use in rolling context windows (N=5 previous segments) rather than full detailed video segment descriptions. We also did context engineering and selected only important features for classification which are only the textual description, topic context, and previous segment summaries, discarding raw visual features after the description generation stage.

These reduction techniques enabled processing of hour-long videos on consumer-grade hardware without sacrificing classification accuracy.

4.3.6 Summary of Preprocessed Data

After completing all the cleaning, transformation, and reduction steps described above, we arrived at two final datasets ready for use in our experiments.

Benchmark Dataset Statistics

The benchmark dataset, summarized in Table and Figure 4.7, consists of 20 educational videos totaling just over 9 hours of content. These videos vary significantly in length the shortest is around 3 minutes while the longest exceeds 1 hour and 20 minutes. This diversity was intentional, as we wanted to test whether our system could handle both small tutorials and full-length lecture recordings. On average, each video contains about 234 segments after our sentence-level segmentation process, though this number ranges widely from 27 segments in shorter videos to nearly 600 in the longest ones.

Metric	Value
Number of Videos	20
Video Duration	
Total Duration	09:13:47
Average Duration	00:27:41
Minimum Duration	00:03:04
Maximum Duration	01:22:15
Segment Count	
Total Segments	4,672
Average Segments	233.6
Minimum Segments	27
Maximum Segments	594

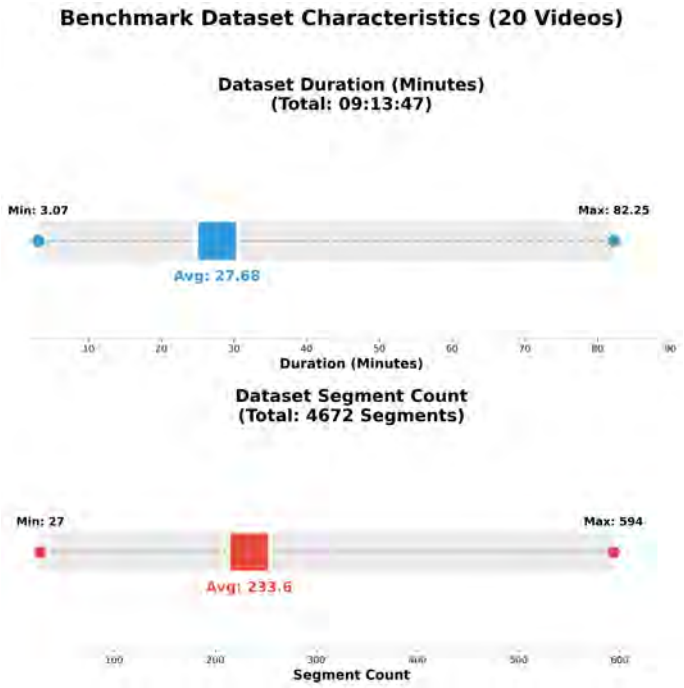

Figure 4.7: Benchmark Dataset Statistics and Composition: (Left) Summary of video metrics and segment counts; (Right) Distribution of durations and counts across 20 educational videos.

Experimental Dataset Statistics

The experimental dataset, shown in Table and Figure 4.8, was used for our fine-tuning experiments on the classification models. We ran two separate experiments: the first used a balanced dataset of roughly 3,000 samples with an almost equal split between relevant and irrelevant labels. For the second experiment, we augmented the data with semi-synthetic samples and hard negatives, resulting in over 7,600 samples split across training (80%), validation (10%), and test (10%) sets. The class distribution remained relatively balanced, with irrelevant samples slightly outnumbering relevant ones at around 54% to 46%.

Table and Figure 4.8 shows the visual breakdowns of fine-tuning dataset history and statistics, Also, showing the distribution of data across different splits.

Split	**Total Samples**	**Relevant**	**Irrelevant**	**Percentage**
Experiment 1 (Balanced Dataset)				
Total	3,064	1,536 (50.13%)	1,528 (49.87%)	100%
Experiment 2 (Augmented Dataset)				
Training	6,114	2,810 (45.96%)	3,304 (54.04%)	80%
Validation	764	351 (45.94%)	413 (54.06%)	10%
Test	764	351 (45.94%)	413 (54.06%)	10%
Total	**7,642**	**3,512**	**4,130**	**100%**

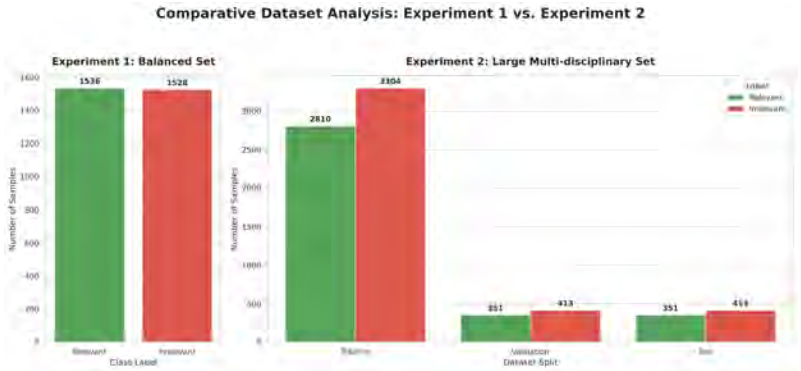

Figure 4.8: Experimental Dataset Statistics and Distribution: Summary table of data point counts (top) and visual bar chart comparison of classes across different splits (bottom).

4.4 Implementation of Selected Design

The proposed orchestration system was implemented in pure python and the system designed as a modular components. All the model(LLM & VLM) inference ran locally using llama.cpp inference engine. This design decouples the model inference from the core pipeline logic.Which allows flexible model replacement using only the user code without touching the core package code. Also, prompts or instructions were also loaded from external files so we can change them easily without modifying any code. As we have used llama.cpp which uses layer offloading and uses quantized GGUF format models to ensure efficient inference on consumer grade CPU and GPU. For the VLM (Qwen3-VL)[31] we have used 8-bit quantized model as vision task is memory heavy. For the language model (Qwen3-4B-Instruct-2507)[41] we have used 16-bit precision to preserve classification accuracy as much as possible. This different precision on different models setup provided a balance between memory efficiency and reasoning accuracy.

The main processing pipeline was implemented as a Python client that communicates with the `llama.cpp` HTTP server for inference requests. Each stage of the pipeline for subtitle processing, video segmentation, frame sampling, description generation, summarization, and classification was implemented as an independent module. This modular structure enables users to swap models, adjust prompts, or adapt the system to different domains or use cases with minimal code changes.

Subtitle generation and alignment were also performed locally. Sentence-level subtitles were generated using the Parakeet TDT 0.6B ASR[42] model, and non-speech events were merged from auxiliary subtitle sources when required. For most experiments, all inference stages were executed on an NVIDIA RTX 3060 GPU with 12 GB VRAM. For a limited number of extremely long videos requiring higher memory capacity, an RTX 5090 GPU was used to complete subtitle inference.

Intermediate and final outputs were stored in lightweight, human-readable formats. Segment-level metadata, timestamps, and classification labels were stored in CSV files for easy inspection and post-processing, while structured representations such as segment descriptions, summaries, and contextual metadata were stored in JSON format. This separation simplifies debugging, evaluation, and downstream integration.

Additionally, as we store all the video description and summary. So using vector embedding-based retrieval system we can do semantic search over video segments. Segment-level textual descriptions were converted into dense vector embeddings using the Qwen3 embedding model [46] and stored in a local vector database. During inference, we can use the embedding of a user query to perform similarity search against all the stored video description embedding on a vector database. The system returns the most relevant video segments based on embedding similarity, along with their corresponding timestamps.

Overall, our implementation prioritized local execution, easy reproducibility and future extensibility. By using quantized model inference with `llama.cpp` with modular Python orchestration system in a low spec hardware, the system achieves efficient and scalable long educational video processing and understanding in a realistic learning environment.
Thesis work link: edu-cut GitHub repository **https://github.com/znydd/edu-cut**

Chapter 5

Result Analysis

5.1 Performance Evaluation

This section discusses the comprehensive result analysis of our proposed framework on detecting irrelevant or off-topic segments in educational videos. This evaluation analysis uses all the key metrics to assess the overall system performance. Along with we discuss the testing methodology to evaluate the system. We also compare performance matrices with other approaches.

5.1.1 Evaluation Metrics

As our system at the end a binary classification system, so the performance of this classification system is evaluated using a confusion matrix as usual, where TP (True Positives), TN (True Negatives), FP (False Positives), and FN (False Negatives) are the four possible classification outcomes in our system. As, our main goal is to identify irrelevant segments, so in our context, a positive classification means *irrelevant* segment, and a negative classification means *relevant* segment.

- **Accuracy:** It shows the overall correctness of the systems classified result across all the video segments. Accuracy provides a general overall indication of systems reliability. But it can be misleading in some case when the dataset is imbalanced.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{5.1}$$

- **Precision (Reliability):** It indicates the proportion of video segments were classified as irrelevant and they are actually irrelevant. A high precision value tells us that relevant video segment is not being removed by misclassification and it is critical for preserving the coherence and integrity of video lectures.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{5.2}$$

- **Recall (Sensitivity):** It measures the system's capability to classify all irrelevant video segments present in a video. High recall score indicates very good and effective ability of detecting irrelevant and off-topic segments, which maximizes the filtering of unnecessary segments.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{5.3}$$

- **F1-Score:** This is the harmonic mean of Precision and Recall, which provides a balanced measure that considers the trade-off between content inclusion and noise reduction. F1-Score is particularly important when both false positives and false negatives are very important.

$$F_1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{5.4}$$

5.1.2 Testing Methodology

The testing methodology we have used is a very rigorous peer-reviewed testing approach which ensure comprehensive and unbiased judgement of overall system performance.

Ground Truth Validation

All model predictions are validated against a benchmark dataset which were human labeled and comprises 20 diverse educational videos with a total duration of more than 9 hours. Each video segment was manually labeled as relevant or irrelevant label by original video consumers and domain experts. With a secondary review process to resolve ambiguous edge cases, which ensures labeling consistency and correctness.

Temporal Grouping

To make the model predictions work with the practical video editing environment, we merged adjacent segments with same classification labels into a continuous group or blocks of timestamps and segments. This post-processing technique makes the resulting cut points usually correspond to the natural segment boundaries of the video. At the end it produces coherent filtered segment output which are suitable for students.

Averaging Techniques

For classification result exploration and deep analysis we have used two complementary averaging strategies to provide a comprehensive picture of system performance:

- **Micro-Averaging:** This approach aggregates all video segments across the entire dataset into a single pool of segments before calculating metrics. So it measures the global system performance. But it is sensitive to the overall distribution of the segments.
- **Macro-Averaging:** Calculates metrics for each video individually and then average them across all video segments. This approach treats each video equally and gives importance to each video regardless of video length and by this it guards against bias from videos with disproportionately high number of segments.

5.2 Analysis of Design Solutions

We have evaluated three distinct approaches systematically to solve the challenge of detecting irrelevant segments in educational videos. Each approach was evaluated using same evaluation criteria with particular importance to accuracy, computational efficiency and feasibility for running on consumer grade hardware.

5.2.1 Gemini-3 Baseline

The Gemini-3 Pro[32] model was evaluated using one-shot strategy, just with raw video input without any subtitle or transcript. So it solely relied on the model's native video understanding and omnimodal capabilities. This baseline establishes a reference for no-effort, state-of-the-art proprietary models without any harness.

Performance Analysis:
Gemini-3 Pro[32] achieved a very good overall accuracy of **0.8163**. However, its precision was notably low at **0.4598**, which indicates aggressive over-prediction of irrelevant segments. This misclassification resulted in the removal of relevant educational content, with nearly half of all predicted irrelevant segments being relevant or false positives. Even though we did not try to increase its capability by using few-shot approach because we are just using it for a baseline and it is gigantic proprietary model.

Feasibility Assessment:
As a proprietary, closed-source, cloud-based solution Gemini-3 presents several limitations(in 1 shot settings):

Cost: To process long-form educational videos it needs a huge API cost, which makes large scale deployment economically non-viable.

Privacy: As is cloud based so we have to upload video content to external servers, which raises privacy concerns for students and educational institutions handling sensitive course materials.

Reproducibility: Proprietary model updates fequently so the outcomes may alter without any notice and impossible to ensure reproducibility.

5.2.2 Experimental Fine-Tuning Approach

We also did an experimental approach where we have utilized LoRA[12] supervised-fine-tuning(SFT) with rank=8,16 but rank=8 had a stable run. So, we did a fine-tuning on Qwen3-4B-instruct-2507[41] model and used it as the classifier in the full orchestration pipeline. The model was trained specifically for irrelevant scenario detection but we had around same amount of relevant scenario data points as irrelevant scenario data points. We increased the irrelevant data points because the loss were decreasing very fast but even after increasing irrelevant segments it showed similar behavior.

Performance Analysis:
This configuration achieved the highest recall among all tested methods (0.9873), successfully identifying nearly all off-topic segments. However, this aggressive de-

tection came at the cost of extremely low precision (0.2783), resulting in an overall accuracy of only 0.5659.

Failure Analysis:
Despite training on a nearly balanced dataset (approximately 50% relevant, 50% irrelevant), the fine-tuned model didn't show any improvement and then when increased the number of irrelevant samples it exhibited severe over-classification toward irrelevant labels. Analysis suggests that the fine-tuning objective inadvertently encouraged the model to treat borderline cases as irrelevant, prioritizing recall at the expense of precision.The resulting output videos suffered from excessive content removal, degrading educational value by eliminating substantial instructional material. Also, from the loss curve we have seen that the model already detected all the training data almost correctly which lead to the loss curve falling close to zero very quickly. So when we increased irrelevant segments amount the model took shortcut and minimal irrelevant segment trigger was enough to classify segments as irrelevant.

5.2.3 Proposed Solution: Orchestration Pipeline

The final proposed system uses the vanilla Qwen3-4B-Instruct-2507[41] model with default weights within our multi-stage orchestration pipeline. This approach leverages the model's inherent instruction following and reasoning capabilities with carefully crafted instructions, context engineering and scalability hacks to process long educational videos.

Performance Analysis:
Our system achieved the optimal balanced performance across all metrics:

Metric Type	Precision	Recall	F1-Score	Accuracy
Micro-averaged	0.8904	0.9277	0.9086	0.9685
Macro-averaged	0.9041	0.9101	0.8973	0.9714

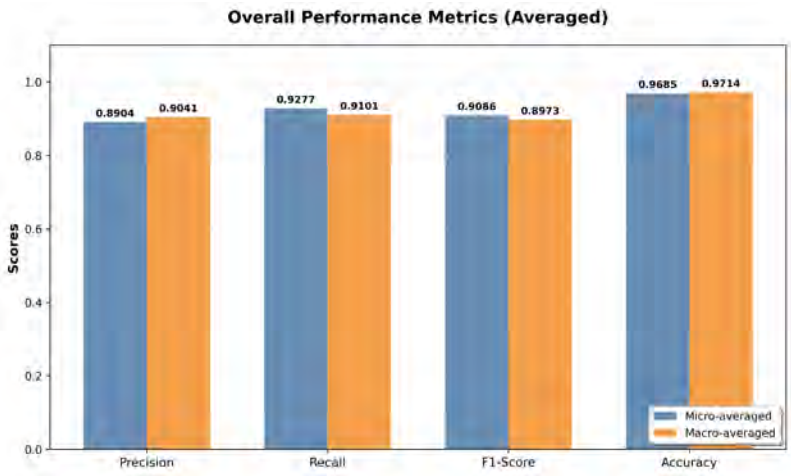

Figure 5.1: Proposed Orchestration System Performance: Summary table of overall performance metrics (top) and bar chart of overall performance metrics (bottom).

- **Accuracy:** 0.9685 — the highest among all tested approaches
- **Precision:** 0.8904 — ensuring minimal removal of educational content
- **Recall:** 0.9277 — effective detection of irrelevant segments
- **F1-Score:** 0.9086 — strong overall performance balance

Efficiency Assessment:
The whole system operates efficiently on consumer-grade hardware or in a single GPU machine(NVIDIA RTX 3060 with 12GB VRAM) and processing educational videos of arbitrary length by dynamic video segmentation. Unlike the fine-tuned classifier, the defalut model demonstrates robust generalization across diverse video types and presentation styles which ensures stable and reliable filtering decisions overall.

5.3 Final Design Adjustments

Based on comprehensive result and performance analysis of initial experiments and iterative development we made several major modifications in our design to enhance system robustness, reliability and overall classification accuracy.

5.3.1 Separation of Description and Classification

The initial design we attempted was to perform both visual understanding and irrelevant scenario detection with a sequential single pass to the Vision Language Model (VLM). But empirical evaluation revealed that small sized VLM hallucinate and struggle to maintain reliable performance while processing visual content and executing complex irrelevance and relevance reasoning at the same time.

Solution:
Our orchestration pipeline went for a two-stage architecture:

1. **Stage 1 (Visual Understanding):** The VLM(Qwen3-VL-4B) generates structured textual descriptions of each video segment. Where it focuses mainly on content summarization, activities, environment understanding without making any classification judgment.

2. **Stage 2 (Relevance Classification):** A dedicated Large Language Model (Qwen3-4B-Instruct-2507) act as a classifier and perform irrelevance and relevance classification based on the textual description, subtitle and rolling context enabling focused reasoning with full context awareness without the visual token load.

This separation improved classification stability by allowing each model to focus on a single well defined task instead of hallucinating while managing all at once.

5.3.2 Transition to Dynamic Segmentation

Early experiments used fixed-length overlapping video segmentation, which frequently cut sentences in middle and messed up the semantic coherence. Mid cut segments often lacked sufficient context for accurate classification. Also, overlapping window is not very useful and inefficient.

Solution:
Fixed overlapping video segmentation was replaced with dynamic, sentence-level segmentation approach using the Parakeet TDT 0.6B[42] automatic speech recognition(ASR) model. Video segments are now defined by natural linguistic boundaries of the spokent audio content. Now each unit or segment represents a complete and coherent semantic element. This approach preserves the full meaning of every little spoken content and enables more accurate classification.

5.3.3 Incorporation of Rolling Context

In a long same topic video, processing isolated video segments for video understanding without temporal context led to hallucination because context window is the whole world for VLM or LLM. This led to wrong and inconsistent classification, particularly for segments whose irrelevance and relevance depends on preceding contents of a video.

Solution:
A rolling context mechanism provides each segment preceding context through summarized descriptions of previous N ($N = 5$) segments. This injected contextual window enables the VLM and LLM classifier to understand chronological flow of the video. Which ultimatly maintain topic continuity across the video segments descriptions, substantially reducing the overall misclassification of context-dependent irrelevant or relevant segments.

5.3.4 Removal of Fine-Tuning Strategy

Although the initial development plan included fine-tuning a dedicated classification model, empirical results demonstrated that fine-tuning produced overly aggressive filtering behavior with unacceptable precision degradation.

Solution:
Our fine-tuning approach was finally abandoned in favor of the default Qwen3-4B-Instruct-2507 model. The model's superior post-training made its generalization capabilities very good as if it behaves like a world model. Also, it was pre-trained with very good condensed dataset, enhanced its world understanding significantly which led to better performance when integrated with our orchestration pipeline. This decision also simplified local usage and ensured reproducibility across different hardware configurations as it is available on HuggingFace.

5.4 Statistical Analysis

This section discusses detailed quantitative results for each evaluated approaches.

5.4.1 Gemini-3 Baseline Results

The Gemini-3 Pro[32] baseline evaluation metrics shows the raw capability of a state-of-the-art proprietary model using one-shot, native video input without external harness. The results indicate moderate accuracy even with huge messy data like video but the precision is not great.

Metric Type	Precision	Recall	F1-Score	Accuracy
Micro-averaged	0.4598	0.5904	0.5170	0.8163
Macro-averaged	0.4867	0.5577	0.4652	0.8127

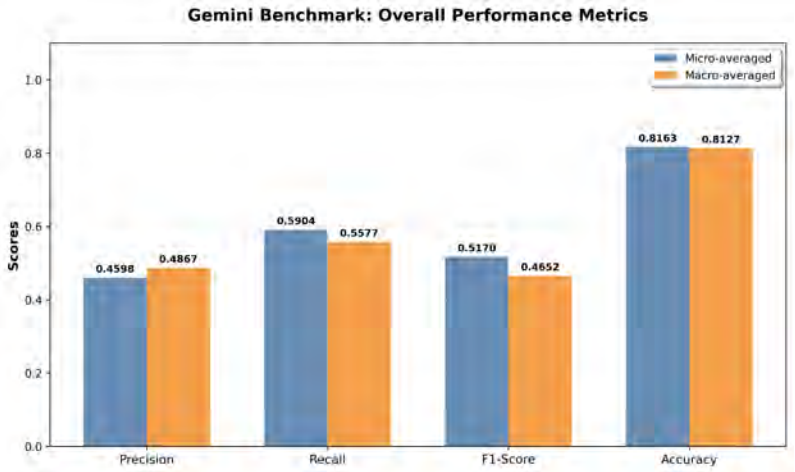

Figure 5.2: Gemini-3 Pro Baseline Metrics: Summary Table of overall performance metrics (top) and visual bar chart of results (bottom).

The low precision 0.4598 demonstrates that even advanced state-of-the-art proprietary models produce substantial false positives when operating without task-specific optimization and native omni-modal input.

5.4.2 Proposed Solution Results

Our multi-staged orchestration system combining VLM description help with LLM classification achieves superior performance across all mentioned evaluated metrics.

Metric Type	Precision	Recall	F1-Score	Accuracy
Micro-averaged	0.8904	0.9277	0.9086	0.9685
Macro-averaged	0.9041	0.9101	0.8973	0.9714

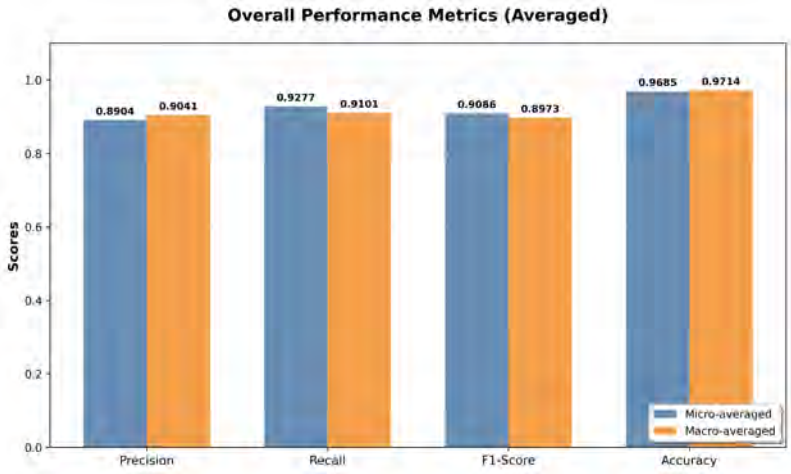

Figure 5.3: Proposed Solution Performance Results: Detailed metric breakdown (top) and comprehensive results comparison (bottom).

The high precision **0.8904** ensures that preservation of relevant educational segment is one of the core priority, while very strong recall score **0.9277** tells us that effective removal of irrelevant and off-topic segments.

5.4.3 Experimental Fine-Tuning Results

The experimental LoRA fine-tuned model demonstrates the disadvantages of aggressive task-specific optimization without reasoning capability. Which led to classification shortcut and skip environment understanding.

Metric Type	Precision	Recall	F1-Score	Accuracy
Micro-averaged	0.2783	0.9873	0.4342	0.5659
Macro-averaged	0.3592	0.9839	0.4877	0.6414

Table 5.1: LoRA Fine-tuning Overall Performance Metrics

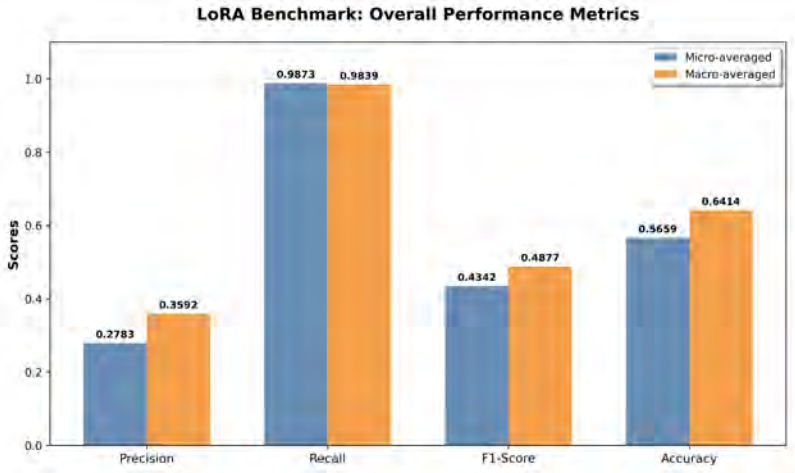

Figure 5.4: LoRA Fine-tuning Visual Performance Results

While achieving exceptional recall score 0.9873, the dangerously low precision score 0.2783 indicates that this approach is unsuitable for practical use educational contexts.

5.5 Comparisons and Relationships

This section presents a structured comparison of all evaluated methods, examines their comparative effectiveness and discusses the key factors influencing variations in performance.

5.5.1 Comparison with Existing Works

The Gemini-3 baseline reflects what is currently considered state-of-the-art performance in proprietary video understanding systems. Although we assume the model uses a complex architecture with a very large number of parameters, its performance remains only moderate when applied to educational video filtering tasks without task-specific adaptation (Accuracy: 0.8163, Precision: 0.4598). This observation indicates that general purpose omni-modal models, even at large scale depend on additional task specific help such as transcripts or topic metadata to accurately differentiate between relevant and irrelevant educational content.

5.5.2 Comparison with Alternative Approaches

In the conducted experiments, the LoRA fine-tuning approach produced the highest recall value (0.9873), which shows that most irrelevant segments were detected. This gain in recall however, coincided with reduced precision (0.2783) and a lower overall accuracy score (0.5659). Such behavior can be attributed to the use of aggressive detection-focused training objective, which appears to shift the model toward labeling uncertain segments as irrelevant, even when some educational information is present. So the model did not reason just got triggered and classified.

5.5.3 Relative Effectiveness Analysis

The orchestration-based method performs better than both the baseline and fine-tuned models, as reflected by the highest accuracy (**0.9685**) and precision (**0.8904**), while maintaining a recall of **0.9277**. Compared to the baseline system, precision increases by 93.7%, rising from 0.4598 to 0.8904, which leads to a clear reduction in false positive predictions and improves the handling of educational content. The fine-tuned model achieves a slightly higher recall (0.9873), whereas the proposed method shows a more even spread across accuracy (71.2% higher), precision (220% higher) and recall. Based on these results, incorporating context-aware engineering appears more suitable for generalization in this task than relying on fine-tuning which is just weight changing or weight manipulation.

We can see the comparison table and figure below 5.2 and 5.5.

5.5.4 Summary Comparison

Approach	Accuracy	Precision	Recall	F1-Score
Gemini-3 Baseline	0.8163	0.4598	0.5904	0.5170
Experimental Fine-Tuned	0.5659	0.2783	**0.9873**	0.4342
Proposed Solution	**0.9685**	**0.8904**	0.9277	**0.9086**

Table 5.2: Comparative Performance Analysis Summary

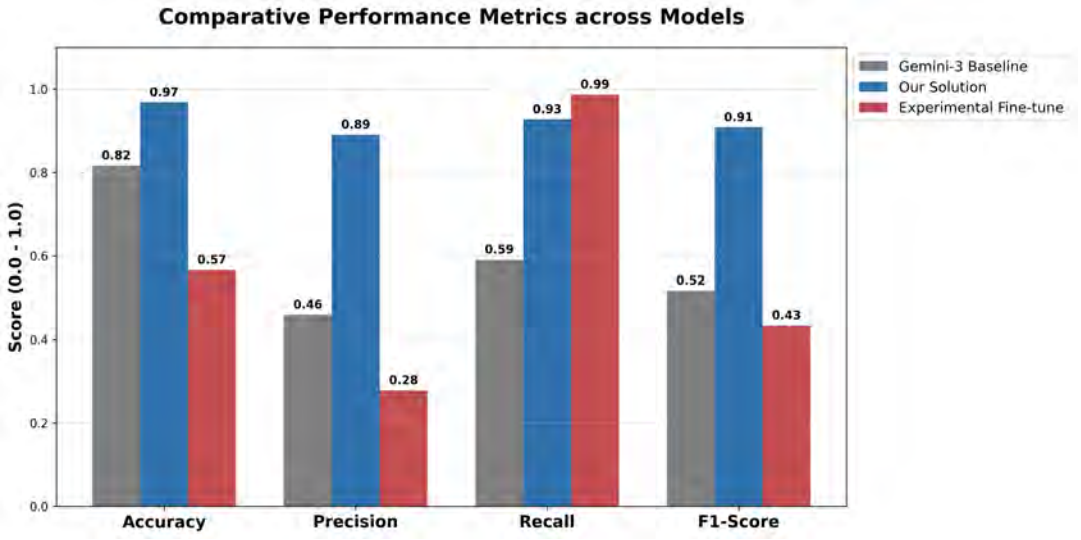

Figure 5.5: Performance Metrics Comparison Across All Evaluated Approaches

Figure 5.6 shows the confusion matrices of the three evaluated models and provides a basis for examining their classification outcomes. In the case of the Gemini-3 baseline, a large number of false positives are observed in the confusion matrix which indicates overly aggressive filtering of content. On the other side, the fine-tuned model reduces false negatives but also increases the number of false positives which results in a stricter filtering outcome.

In comparison with those two approaches, the proposed method produces predictions that are more evenly distributed, with fewer false positives and fewer false negatives overall. The stronger diagonal concentration in its confusion matrix suggests more reliable classification and improved retention of relevant educational material while filtering out irrelevant segments.

Figure 5.6: Confusion Matrix Comparison: (Left) Gemini-3 Baseline, (Center) Proposed Solution, (Right) Experimental Fine-Tuned Model

5.6 Discussions

5.6.1 Interpretation of Results

From the experimental result it can be said that our proposed two-stage method is quite better than the one-stage system. The system can control each single task more efficiently. The experimental result reflects that small AI based model works better when a large complex task is being broken down into smaller and precisely taken steps rather than doing a task all at once. From the success of the orchestration pipeline, it is quite visible that through good context design the limitation of small models can be reduced. Structured context is being provided by the system, that helps the model to understand lengthy videos more distinctly. Through this the system can behave in such a way a human naturally applies while watching the educational videos.

5.6.2 Implications for Educational Technology

Our proposed system has important outcomes for improving the access to the educational videos. This system allows constructive filtering on videos on consumer-grade hardware, through this the technical and the financial barriers are being removed for each student, educators and institution. Therefore advanced video analysis has become more accessible. The framework carry the privacy-safety analysis through performing all the processing locally. This means the confidential educational videos no need to be uploaded to the cloud services which keep the privacy in academic environments. Also the system is scalable and can work on any length videos. Then make sure that the framework can be used for a broad range of learning environments.

5.6.3 Limitations

Several limitations of the current work are:

Language Constraint: The current system is mainly designed for the English-Language educational contents. The Automatic Speech Recognition(ASR) model (Parakeet TDT 0.06B)[42] and the VLM and LLM are also good at English only. This issue has created a limitation for the use of the system for non-English contents unless the additional adaptation is being performed.

Audio Quality Dependence: System performance relies on accurate speech recognition. Videos with heavy background noise, strong accents, or overlapping speech may produce degraded transcripts, potentially affecting classification accuracy.

Subjectivity of Relevance: The definition of "irrelevant" content is inherently subjective. Content considered off-topic by some learners (e.g., personal anecdotes, historical context) may be valuable to others. The current system targets only clearly irrelevant content (silence, technical difficulties, administrative announcements) to minimize controversial decisions.

Silent Visual Instruction: The audio-first classification approach may incorrectly flag silent demonstrations (e.g., mathematical derivations on a whiteboard) as irrelevant. However, such content is typically followed by verbal explanation, which preserves the instructional material in adjacent segments.

5.6.4 Future Work

There are several directions for future research that emerge from this work. First adding multilingual educational content(specially Bengali) through multi-lingual ASR model and LLM integration Then making the benchmark dataset more robust, diverse and larger. Also, exploration of segment-level content retrieval and question-answering capabilities using the generated textual descriptions

Chapter 6

Conclusion

6.1 Summary of Findings

Through the study it can be said that the two-stage method for filtering educational videos is the best way. Here first through the VLM a text based description of each educational video being created, after that a LLM decides which part of the video is useful and which part is irrelevant. We have used the Qwen3-4B model here. Our tests also reflects that the dynamic segmentation, through which the videos being cut based on punctuation rather than the fix timers, is way better for keeping the lecture videos meaningful and irrelevant part less.

6.2 Contributions to the Field

Our system can be used by the fully open-source tools which can be work on the home computers. By this our system allows students and teachers process lengthy videos without using the expensive cloud service. Through our system it can be said that a properly video filtering is possible by using small and systematic AI models.

Additionally, we have curated a benchmark dataset of 20 educational videos with manually annotated ground truth labels, which can serve as a standardized evaluation resource for future research in video content filtering.

6.3 Recommendations for Future Work

We have seen that other benchmarking datasets such as TVSum or SumMe[4], [5] have 25-50 videos. So we also curated a benchmarking dataset of 20 videos for benchmarking. We intend to add more videos to our dataset, from various other domains to make the dataset more diverse, and publish a more robust benchmarking dataset which can be used by anyone to test models respective to similar tasks like ours. Then we also intend to complete the implementation of content retrieval in our system with the help of the local vector database, and create a benchmarking dataset for this task as well. Furthermore, extending the system to support multi-language content would significantly broaden its applicability, allowing non-English educational videos to be processed effectively.

Bibliography

[1] R. Socher and L. Fei-Fei, "Connecting modalities: Semi-supervised segmentation and annotation of images using unaligned text corpora," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2010, pp. 966–973. DOI: 10.1109/CVPR.2010.5540112

[2] J. Weston, S. Bengio, and N. Usunier, "Large scale image annotation: Learning to rank with joint word-image embeddings," *Machine Learning*, vol. 81, no. 1, pp. 21–35, 2010. DOI: 10.1007/s10994-010-5198-3 [Online]. Available: https://doi.org/10.1007/s10994-010-5198-3

[3] C. Zauner, "Implementation and benchmarking of perceptual image hash functions," Diploma Thesis, University of Applied Sciences Upper Austria, Hagenberg Campus, 2010. [Online]. Available: https://www.phash.org/docs/pubs/thesis_zauner.pdf

[4] M. Gygli, H. Grabner, H. Riemenschneider, and L. Van Gool, "Creating summaries from user videos," in *Proceedings of the European Conference on Computer Vision (ECCV)*, Springer, 2014, pp. 505–520.

[5] Y. Song, J. Vallmitjana, A. Stent, and A. Jaimes, "Tvsum: Summarizing web videos using titles," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 5179–5187.

[6] E. Péter and H. Ferenc, "Analysis of video views in online courses," in *2017 40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 2017, pp. 778–782. DOI: 10.23919/MIPRO.2017.7973527

[7] M. Rochan and Y. Wang, "Video summarization by learning from unpaired data," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2019, pp. 7902–7911.

[8] J. Kaplan et al., "Scaling laws for neural language models," *arXiv preprint arXiv:2001.08361*, 2020.

[9] A. Sarkar, S. Megha, P. Srilekha, and S. Niveditha, "Video summarization using subtitles," *International Journal of Emerging Technologies and Innovative Research (JETIR)*, vol. 7, no. 5, pp. 5–8, May 2020. [Online]. Available: https://www.jetir.org/papers/JETIRDV06002.pdf

[10] A. Radford et al., "Learning transferable visual models from natural language supervision," in *Proceedings of the 38th International Conference on Machine Learning*, M. Meila and T. Zhang, Eds., ser. Proceedings of Machine Learning Research, vol. 139, PMLR, 18–24 Jul 2021, pp. 8748–8763. [Online]. Available: https://proceedings.mlr.press/v139/radford21a.html

[11] J. Hoffmann, S. Borgeaud, A. Mensch, E. Buchatskaya, T. Cai, E. Rutherford, et al., "Training compute-optimal large language models," in *Advances in Neural Information Processing Systems*, vol. 35, 2022, pp. 30 016–30 030.

[12] E. J. Hu et al., "LoRA: Low-rank adaptation of large language models," in *International Conference on Learning Representations*, 2022. [Online]. Available: https://openreview.net/forum?id=nZeVKeeFYf9

[13] S. Lin, J. Hilton, and O. Evans, "TruthfulQA: Measuring how models mimic human falsehoods," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 3214–3252. DOI: 10.18653/v1/2022.acl-long.229 [Online]. Available: https://aclanthology.org/2022.acl-long.229

[14] J. Huang and K. C.-C. Chang, "Towards reasoning in large language models: A survey," in *Findings of the Association for Computational Linguistics: ACL 2023*, Toronto, Canada: Association for Computational Linguistics, Jul. 2023, pp. 1049–1065. DOI: 10.18653/v1/2023.findings-acl.67 [Online]. Available: https://aclanthology.org/2023.findings-acl.67

[15] H. Li, Q. Ke, M. Gong, and T. Drummond, "Progressive video summarization via multimodal self-supervised learning," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, IEEE, 2023, pp. 5584–5593. DOI: 10.1109/WACV56688.2023.00554

[16] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," in *Proceedings of the 40th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 202, PMLR, 2023, pp. 28 492–28 518. [Online]. Available: https://proceedings.mlr.press/v202/radford23a.html

[17] S. Samsi et al., "From words to watts: Benchmarking the energy costs of large language model inference," in *2023 IEEE High Performance Extreme Computing Conference (HPEC)*, 2023, pp. 1–9. DOI: 10.1109/HPEC58863.2023.10363447

[18] H. Touvron et al., *Llama 2: Open foundation and fine-tuned chat models*, 2023. arXiv: 2307.09288 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2307.09288

[19] Anthropic, "The claude 3 model family: Opus, sonnet, haiku," Anthropic, Tech. Rep., Mar. 2024. [Online]. Available: https://www-cdn.anthropic.com/de8ba9b01c9ab7cbabf5c33b80b7bbc618857627/Model_Card_Claude_3.pdf

[20] D. M. Argaw et al., "Scaling up video summarization pretraining with large language models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2024, pp. 8332–8341.

[21] Z. Chen et al., "Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 24 185–24 198.

[22] A. Gholami, Z. Yao, S. Kim, C. Hooper, M. W. Mahoney, and K. Keutzer, "Ai and memory wall," *IEEE Micro*, vol. 44, no. 3, pp. 33–39, 2024. DOI: 10.1109/MM.2024.3373763

[23] C.-P. Hsieh et al., "RULER: What's the real context size of your long-context language models?" In *First Conference on Language Modeling*, 2024. [Online]. Available: https://openreview.net/forum?id=kIoBbc76Sy

[24] N. F. Liu et al., "Lost in the middle: How language models use long contexts," *Transactions of the Association for Computational Linguistics*, vol. 12, pp. 157–173, 2024. DOI: 10.1162/tacl_a_00638 [Online]. Available: https://aclanthology.org/2024.tacl-1.9/

[25] OpenAI et al., *Gpt-4 technical report*, 2024. arXiv: 2303.08774 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2303.08774

[26] D. Rein et al., "GPQA: A graduate-level google-proof q&a benchmark," in *First Conference on Language Modeling*, 2024. [Online]. Available: https://openreview.net/forum?id=Ti67584b98

[27] G. Team et al., *Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context*, 2024. arXiv: 2403.05530 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2403.05530

[28] Y. Wang et al., "Mmlu-pro: A more robust and challenging multi-task language understanding benchmark," in *Advances in Neural Information Processing Systems*, A. Globerson et al., Eds., vol. 37, Curran Associates, Inc., 2024, pp. 95 266–95 290. DOI: 10.52202/079017-3018 [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2024/file/ad236edc564f3e3156e1b2feafb99a24-Paper-Datasets_and_Benchmarks_Track.pdf

[29] Z. Allen-Zhu and Y. Li, "Physics of Language Models: Part 3.3, Knowledge Capacity Scaling Laws," in *Proceedings of the 13th International Conference on Learning Representations*, ser. ICLR '25, Full version available at https://ssrn.com/abstract=5250617, Apr. 2025.

[30] S. Bai et al., *Qwen2.5-vl technical report*, 2025. arXiv: 2502.13923 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2502.13923

[31] S. Bai et al., "Qwen3-vl technical report," *arXiv preprint arXiv:2511.21631*, 2025.

[32] G. DeepMind, "Gemini 3 pro model card," Google DeepMind, Tech. Rep., Nov. 2025. [Online]. Available: https://storage.googleapis.com/deepmind-media/Model-Cards/Gemini-3-Pro-Model-Card.pdf

[33] D. Guo et al., "Deepseek-r1 incentivizes reasoning in llms through reinforcement learning," *Nature*, vol. 645, no. 8081, pp. 633–638, Sep. 2025, ISSN: 1476-4687. DOI: 10.1038/s41586-025-09422-z [Online]. Available: http://dx.doi.org/10.1038/s41586-025-09422-z

[34] J. Heo and S. Lee, "A study on applying large language models to issue classification," in *2025 IEEE/ACM 33rd International Conference on Program Comprehension (ICPC)*, 2025, pp. 1–11. DOI: 10.1109/ICPC66645.2025.00022

[35] L. Huang et al., "A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions," *ACM Trans. Inf. Syst.*, vol. 43, no. 2, Jan. 2025, ISSN: 1046-8188. DOI: 10.1145/3703155 [Online]. Available: https://doi.org/10.1145/3703155

[36] C. Hur et al., "Narrating the video: Boosting text-video retrieval via comprehensive utilization of frame-level captions," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2025, pp. 24 077–24 086.

[37] M. J. Lee, D. Gong, and M. Cho, "Video summarization with large language models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2025, pp. 18 981–18 991.

[38] M. J. Lee, D. Gong, and M. Cho, "Video summarization with large language models," in *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025, pp. 18 981–18 991. DOI: 10.1109/CVPR52734.2025.01768

[39] A. Marafioti et al., "Smolvlm: Redefining small and efficient multimodal models," *arXiv preprint arXiv:2504.05299*, 2025.

[40] OpenAI et al., *Gpt-oss-120b & gpt-oss-20b model card*, 2025. arXiv: 2508.10925 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2508.10925

[41] Qwen Team, "Qwen3 technical report," 2025. arXiv: 2505.09388. [Online]. Available: https://arxiv.org/abs/2505.09388

[42] M. Sekoyan et al., "Canary-1b-v2 & parakeet-tdt-0.6b-v3: Efficient and high-performance models for multilingual asr and ast," 2025. arXiv: 2509.14128 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2509.14128

[43] G. Team et al., *Gemma 3 technical report*, 2025. arXiv: 2503.19786 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2503.19786

[44] K. Team et al., *Kimi k2: Open agentic intelligence*, 2025. arXiv: 2507.20534 [cs.LG]. [Online]. Available: https://arxiv.org/abs/2507.20534

[45] Twelve Labs Team, *Marengo 3.0: Defining real-world embedding ai for videos, audios, texts, images, and composition of them*, https://www.twelvelabs.io/blog/marengo-3-0, Accessed: 2026-01-29, Nov. 2025.

[46] Y. Zhang et al., *Qwen3 embedding: Advancing text embedding and reranking through foundation models*, 2025. arXiv: 2506.05176 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2506.05176

[47] N. Zeng, H. Hou, F. R. Yu, S. Shi, and Y. T. He, "Scenerag: Scene-level retrieval-augmented generation for video understanding," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, To appear, 2026.

Appendix A (Prompts and Output example)

VLM Description Prompt:

You are an expert video-language model. Your task is to analyze the current video clip using:

- The visual frames of this clip
- The transcript/subtitle of this clip
- The global topic of the entire video
- The summaries of the previous 4–5 segments (chronological context)

You must describe **everything** that occurs in the current clip — not just educational content.

You MUST include:

- People, objects, environment, and layout
- Actions, gestures, movements, expressions
- On-screen text, slides, diagrams, graphics
- Irrelevant or off-topic events (noise, mistakes, interruptions)
- Camera angles, transitions, zooms, cuts
- Background motion or audio cues (if indicated in transcript)

Rules

- Use **both** visual and textual evidence.
- Maintain **chronological**, time-ordered flow.
- Do **not** hallucinate anything not visible or mentioned.
- Use previous segment summaries **only for continuity**, not for guessing missing details.
- The output must represent **exactly** what happens in this specific clip.

TASK
Describe the current clip in complete detail.

Focus on:
- Everything visible in every frame
- Everything spoken in the transcript
- Any irrelevant, accidental, or off-topic events
- Physical surroundings, objects, lighting, environment
- All actions from start to end
- Any visual transitions or camera movements

OUTPUT FORMAT

Clip Description (Chronological)
* {Detailed, time-ordered step-by-step description of everything that happens}

Visual Details
* {List of all visible objects, people, environment details}

Notable Irrelevant or Off-Topic Elements
* {List of interruptions, irrelevant actions, noises, mistakes, random events}

Transcript Summary
* {Concise, accurate summary of what is spoken during the clip}

Figure 6.1: Classification Prompt

LLM Classification Prompt:

You are a **Senior Post-Production Editor** specializing in high-impact educational video content. Your mission is to transform raw classroom/webinar footage into a streamlined, "zero-distraction" learning experience.

Your goal is to increase the **Signal-to-Noise Ratio**. In post-production, we cut "the fat" so that future students can focus purely on the pedagogy.

The Golden Rule of the Edit
Audio is the Lead; Visuals are the Support.
In an educational context, "teaching" happens through the explanation. A visual (like a slide or a whiteboard) is "Dead Air" if there is no accompanying pedagogical commentary.

Your Task: Analyze the provided segment and decide if it stays in the final cut (****Relevant****) or gets trimmed (****Irrelevant****).

Decision Logic: The Editor's Cut
1. The "Keep" Criteria (Relevant)
A segment is **Relevant** ONLY if it meets all three:
* **Active Instruction:** The transcript shows the teacher actively explaining, defining, demonstrating, or narrating the lesson.
* **Topic Alignment:** The speech directly advances the "Video Topic."
* **Engagement:** The segment provides clear academic value that a student would need for an exam or practical application.

2. The "Trim" Criteria (Irrelevant) — "When in doubt, cut it out."
A segment is **Irrelevant** if ANY of the following "Production Noise" is detected:
* **Dead Air:** Any silence or non-instructional pause longer than **5 seconds.**
* **The "Visual Fallacy":** A slide, code editor, or whiteboard is visible, but the teacher is silent, shuffling papers, or talking about something else. (If they aren't teaching, the visual doesn't matter).
* **Technical Friction:** Mic checks, "Can you hear me?", screen-sharing lag, software loading, "Let me find my notes," or troubleshooting.
* **Logistics & Housekeeping:** Greetings ("Hi everyone"), goodbyes, attendance, "Is the recording started?", or syllabus/administrative chatter.
* **Social Noise:** Off-topic jokes, personal anecdotes, side-conversations with specific students that don't benefit the general audience, or background environmental noise.
* **Filler/Stalling:** Excessive "umms," "hang on a sec," or repetitive circling while the teacher tries to remember a point.

Input Data Points
To make your editorial decision, you will receive:
1. **Video Topic:** The specific lesson objective.
2. **Chronological Context:** What happened just before this (to maintain flow).
3. **Current Timestamp:** The exact window of the segment.
4. **Transcript:** The **Primary Signal** of whether teaching is occurring.
5. **Segment Description:** VLM-generated visual context (The **Secondary Signal**).

Strict Output Format
You are part of an automated editing pipeline. You must output **only** a valid JSON object. No intro, no outro.

```json
{
"type": "Relevant" or "Irrelevant",
"reasoning": "A concise, professional editorial justification for why this stays or is trimmed."
}
```

Editorial Examples for Your Reference

* **Scenario:** Teacher is showing a complex math formula on screen but is currently looking for their water bottle and saying "One second, I'm parched."
* ****Your Decision:** Irrelevant**. (Visuals exist, but the audio is non-pedagogical noise).

* **Scenario:** Teacher is mid-sentence explaining a concept, but the screen is just a black 'waiting' slide.
* ****Your Decision:** Relevant**. (The transcript contains the teaching value; we can fix the visual in the 'edit,' but the information is vital).

* **Scenario:** Teacher says, "Now, look at this line of code," followed by 8 seconds of silence while they type.
* ****Your Decision:** Irrelevant**. (8 seconds of silence is a distraction; this is a 'trim' point).

Figure 6.2: Classification Prompt

VLM Description Output:

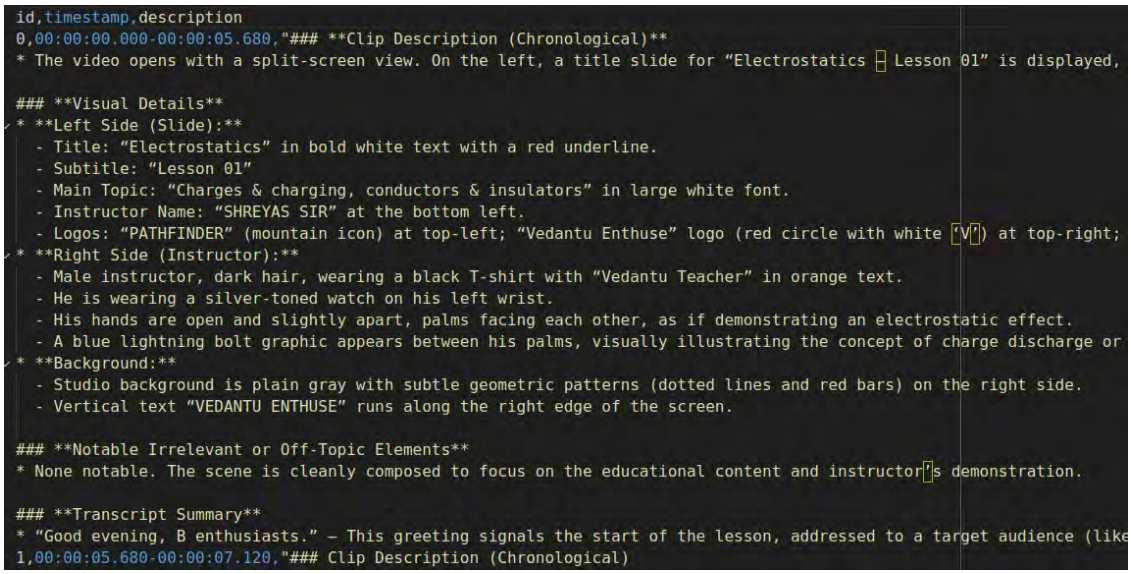

Figure 6.3: Description from VLM

LLM Classification Output:

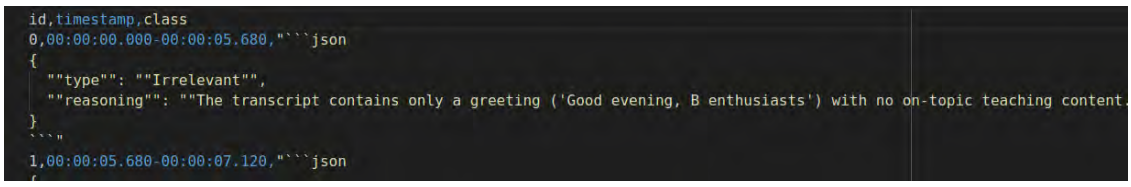

Figure 6.4: Classification result from LLM