Abstractive Summarization of Bengali Academic Videos Based on Audio Subtitles

Lamisa Bintee Mizan Deya[1], **Farhatun Shama**[1], **Abdul Aziz**[2,1],
Md Kaykobad Reza[3], **Md Shahidul Salim**[4,1]
[1]CSE, KUET, Bangladesh [2]CSE, HBKU, Doha, Qatar [3]UC Riverside [4]UMass Lowell
{deya1907049, shama1907033}@stud.kuet.ac.bd abaz89721@hbku.edu.qa
mreza025@ucr.edu mdshahidul_salim@student.uml.edu

Abstract

The rapid growth of academic video content makes it difficult for students and educators to find relevant information efficiently. This is especially challenging for low-resource languages like Bengali due to the lack of a video summarization tool. This paper presents the first end-to-end pipeline for the abstractive summarization of Bengali academic videos. The proposed system preprocesses audio to improve transcription quality and converts speech to text using Google's Speech Recognition API. The text is segmented using a smart chunking method to be compatible with the model's context window. For summarization, we fine-tuned the *BanglaT5* model on a new benchmark dataset of 10,029 text-summary pairs obtained from educational videos. To generate relevant titles, we fine-tuned the *mT5-multilingual-XLSum* model on our curated dataset of 1,005 summary-title pairs. Our fine-tuned summarization model shows strong performance, achieving F1 scores of 0.8793 (BERTScore), 0.3894 (ROUGE-1), and 0.2557 (ROUGE-L), outperforming other models. Our title generation model achieved ROUGE-1 and ROUGE-L F1 scores of 0.4476 and 0.3720, respectively. The summaries include timestamps for easy video navigation. This work aims to improve the accessibility of educational content in Bengali. It also contributes valuable datasets and a robust baseline system that demonstrates strong zero-shot capabilities on other spoken contents. The code and datasets are available at https://github.com/LamisaDeya/Bangla-Video-Summarization.

1 Introduction

The exponential growth of educational video content on platforms like YouTube makes it difficult for students to find relevant content (Mamedova et al., 2023; Apostolidis et al., 2021a). The problem is further extended in the post-pandemic era, where online classes and webinars became common, yet many students miss them due to scheduling or time zone conflicts (Kumar and Kabiri, 2022; Tănase et al., 2022). Although various video summarization methods exist for English (Alrumiah and Al-Shargabi, 2022), no comprehensive solution is available for Bengali educational videos, despite Bengali being spoken by over 300 million people and widely used in education across Bangladesh and West Bengal (Al Maruf et al., 2024).

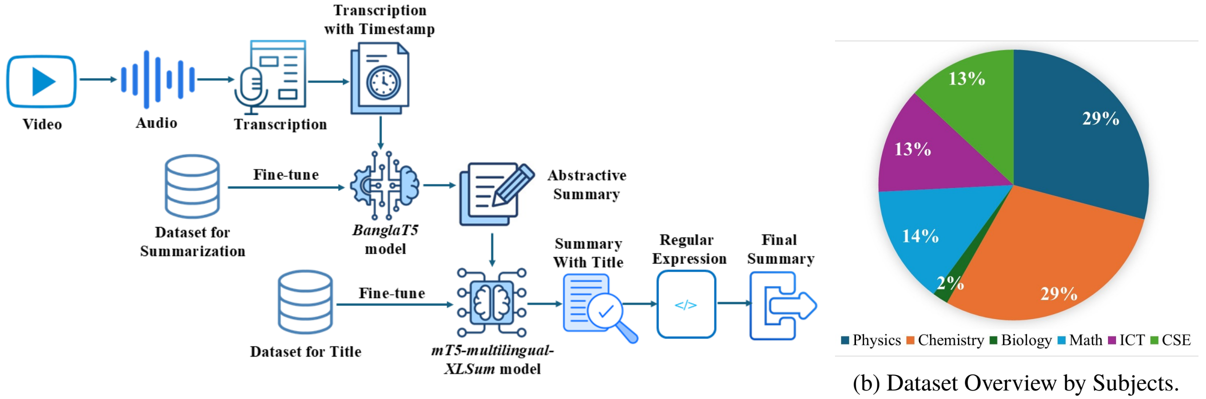
Existing summarization models have been developed mainly on short, formal text like news articles. There is no available system for directly summarizing a Bengali video. Also, existing Bengali text summarization datasets (Hasib et al., 2023; Bhattacharjee et al., 2020) fail to capture the informal, conversational patterns of spoken academic content (Horowitz and Samuels, 2023). So, summarizing informal speech data in Bengali has remained largely unexplored. Moreover, video summarization involves additional challenges, including transcription, appropriate title generation (Yang et al., 2024), and temporal alignment for mapping summary segments back to corresponding video portions (Ying et al., 2024).

Existing Bengali automatic speech recognition (ASR) systems lack accuracy, and so the summary quality is degraded. In addition, since current datasets focus on formal text, they fail to handle spontaneous spoken data. Thus, the quality of summarization in the academic domain has remained low, and important contextual information is often lost. To address these limitations, we propose an end-to-end Bengali video summarization framework illustrated in Figure 1 that integrates speech transcription, summarization, title generation, and integration of timestamps in a single pipeline.

Our key contributions are given below:

- We have created two benchmark datasets for both summarization and title generation.
- We have developed a complete system that in-

Findings of the Association for Computational Linguistics: EACL 2026, pages 6624–6643
March 24-29, 2026 ©2026 Association for Computational Linguistics



(a) Video Summarization Pipeline.

(b) Dataset Overview by Subjects.

Figure 1: Overview of the Summarization Process and Dataset Insights.

tegrates audio pre-processing, context-aware smart chunking, abstractive summarization, and title generation methods for academic videos. *BanglaT5* (bT5) and *mT5-multilingual-XLSum* models significantly outperform existing models for summarization and title generation.

2 Related Works

Prior works in video summarization include a tag-based framework for lecture transcripts (Kim and Kim, 2016) and a visual key frame extraction method using local features and graph clustering (Gharbi et al., 2019). But these approaches were not in Bengali.

Abstractive summarization has progressed from statistical methods to deep learning models, particularly encoder-decoder architectures with attention (Apostolidis et al., 2021b; Zhang et al., 2022). For Bengali, seq2seq models with RNNs/LSTMs and unsupervised methods like BenSumm show strong results (Talukder et al., 2019; Chowdhury et al., 2021). Transformer-based models such as BART and its variants address long documents (Wilman et al., 2024). Multilingual models like mBART and mT5 extend cross-lingual capabilities but face challenges in low-resource settings (Li et al., 2024a; Masih et al., 2025; Dhakal and Baral, 2024; Xue et al., 2021). Specifically, bT5, trained on a large Bengali corpus, outperforms multilingual models and achieves state-of-the-art results in Bengali summarization, with both hybrid and abstractive approaches demonstrating significant improvements (Raffel et al., 2023; Bhattacharjee et al., 2023; Hasib et al., 2023; Hayat et al., 2023).

Recent models on summarization like, Qwen series, built on transformer architectures and pretrained on up to 3 trillion tokens, demonstrate strong multilingual summarization performance (Bai et al., 2023). LLaMA-2's decoder-only transformer architecture offers different trade-offs, while Microsoft's compact Phi-3 Mini model provides efficient on-device deployment options (Touvron et al., 2023; Abdin et al., 2024).

Abstractive methods are preferred over extractive methods due to more natural, human-like summaries (Giarelis et al., 2023; Bhargav et al., 2022). Studies consistently show **monolingual models outperforming multilingual approaches** like mT5 in low-resource settings (Hasan et al., 2021; Bai et al., 2021; Li et al., 2024b).

Title generation research explores both extractive and abstractive methods, with graph-based models learning sentence structures for better headline generation (Zhang et al., 2020). Hybrid approaches combining Convolutional Neural Networks (CNNs) and RNNs with reinforcement learning show improved ROUGE scores (Singh et al., 2021), while transformer-based models like mBART and mT5 demonstrate effectiveness for multilingual headline generation tasks (Bukhtiyarov and Gusev, 2020; MADASU, 2024).

Automatic speech recognition has advanced with self-supervised models like wav2vec 2.0 (Baevski et al., 2020), OpenAI's Whisper trained on 680,000 hours for robust multilingual ASR (Radford et al., 2022), and Google's Universal Speech Model achieving state-of-the-art across 100+ languages using Conformers and BEST-RQ pre-training (Zhang et al., 2023).

Evaluation of summarization and title generation is challenging since traditional metrics

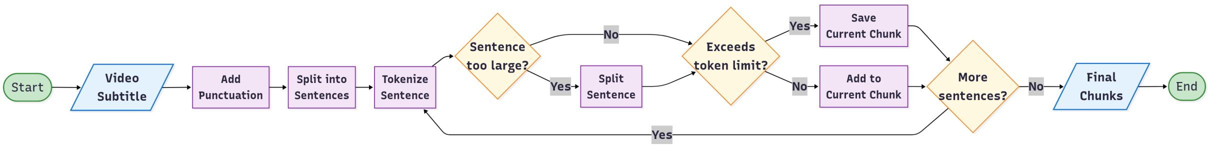


Figure 2: Flowchart of Dataset Preparation for Summarization.

like ROUGE and BLEU fail to capture semantic and factual accuracy (Davoodijam and Alambardar Meybodi, 2024; Deutsch and Roth, 2021). Recent studies highlight reference-free metrics such as BERTScore, MoverScore, and factual consistency measures for better LLM evaluation (Tang et al., 2023; Zhang et al., 2024, 2025). The bootstrap technique estimates metric uncertainty via resampling (Raschka, 2020). A comparative summary of related works is given in Table 15 (Appendix).

3 Methodology

This study follows a systematic methodology comprising data collection, preprocessing, model fine-tuning, and post-processing.

3.1 Data Collection and Dataset Generation

Existing summarization datasets work well for formal text, but video summarization requires a curated dataset for spoken academic content. As shown in Figure 2, processed subtitle chunks from 213 Bengali academic videos (covering 6 subjects and 46 topics) were paired with manually generated summaries to form an Excel dataset. Videos from over 200 speakers ensure dialectal diversity and balanced subject coverage (Figure 1(b)). Additionally, a benchmark dataset of summary–title pairs was created for precise academic title generation in Bengali. The datasets were divided into 80% for training, 10% for validation, and 10% for testing. Table 1 summarizes the key statistics of the datasets.

3.2 Data Pre-processing

To get the subtitle, firstly, we need the audio to be extracted from the video. We achieved this using *MoviePy*, a Python library that separates the audio stream and saves it in *WAV* format after decoding (Razdan et al., 2024).

Transcription quality directly impacts summarization. To improve accuracy, audio was preprocessed (Algorithm 1, Figure 5 in the appendix) and split into 45-second chunks for API compatibility. Each chunk's dBFS levels were analyzed, amplified as needed, and normalized only if peaks exceeded −3 dBFS. Chunk-based amplification was used for efficiency and quality assurance, avoiding distortion from global adjustments or the computational cost of sample-level processing.

Algorithm 1 Audio Preprocessing.

```
Require: Audio file $A$
Ensure: Transcription from amplified audio chunks
Initialize:
$chunk\_size \leftarrow 45$ seconds
$chunks \leftarrow \text{SplitAudio}(A, chunk\_size)$
for each $chunk_i$ in $chunks$ do
    Measure Audio Levels:
    $dBFS_{avg} \leftarrow \text{CalculateAverage}(chunk_i)$
    $dBFS_{peak} \leftarrow \text{CalculatePeak}(chunk_i)$
    Determine Amplification:
    if $dBFS_{avg} < -35$ then
        $boost \leftarrow 8$ dB
    else if $-35 \leq dBFS_{avg} < -25$ then
        $boost \leftarrow 6$ dB
    else if $-25 \leq dBFS_{avg} < -15$ then
        $boost \leftarrow 3$ dB
    else
        $boost \leftarrow 0$ dB
    end if
    Apply Amplification:
    $chunk_i^{amp} \leftarrow \text{ApplyBoost}(chunk_i, boost)$
    $dBFS_{peak}^{new} \leftarrow \text{CalculatePeak}(chunk_i^{amp})$
    Check and Normalize:
    here, N -> Normalization, hr -> headroom
    if $dBFS_{peak}^{new} > -3$ then
        $chunk_i^{final} \leftarrow \text{N}(chunk_i^{amp}, \text{hr} = 3 \text{ dB})$
    else
        $chunk_i^{final} \leftarrow chunk_i^{amp}$
    end if
    Export and Transcribe:
    $temp\_file \leftarrow \text{ExportWAV}(chunk_i^{final})$
    $text_i \leftarrow \text{GoogleSpeechAPI}(temp\_file)$
end for
return Concatenate($\{text_i\}$)
```

Existing transcription methods perform poorly for Bengali. Among seven tested approaches, Google's multilingual *Universal Speech Model (USM)* (Zhang et al., 2023) produced the best results. It recognizes over 100 languages. Preprocessed audio chunks were transcribed via the USM API and concatenated to form the full text.

The transcription does not contain punctuation, but it is necessary to segment subtitles into coherent sentences, which in turn improves fluency

Dataset	Pairs	Max Words	Min Words	Videos	Topics	Playlists
Summarization	10,029	Text: 422 / Sum: 199	Text: 401 / Sum: 150	213	46	18
Title Generation	1,005	Sum: 1,011 / Title: 19	Sum: 1,003 / Title: 7	335	54	18

Table 1: Dataset Statistics for Summarization and Title Generation. Sum denotes Summary.

Algorithm 2 Token-Based Chunking with One-Sentence Overlap.

```
Require: Punctuated text $T$, tokenizer $\tau$, max_token $M$
Ensure: Set of overlapping text chunks $\mathcal{C}$
Initialize: $\mathcal{C} \leftarrow \emptyset, i \leftarrow 1$
here, current_chunk -> cc, token_count -> tc, overlap_sentence -> os
$sentences \leftarrow$ SplitIntoSentences($T$)
while $i \leq |sentences|$ do
    $cc \leftarrow \emptyset, tc \leftarrow 0$
    $os \leftarrow$ null
    Build Current Chunk:
    while $i \leq |sentences|$ do
        $s_i \leftarrow sentences[i]$
        $tokens_i \leftarrow \tau(s_i)$
        if $tc + |tokens_i| \leq M$ then
            $cc \leftarrow cc \cup \{s_i\}$
            $tc \leftarrow tc + |tokens_i|$
            $os \leftarrow s_i$
            $i \leftarrow i + 1$
        else
            break
        end if
    end while
    Finalize Chunk:
    if $cc \neq \emptyset$ then
        $\mathcal{C} \leftarrow \mathcal{C} \cup \{cc\}$
        Set Overlap for Next Chunk:
        if $i \leq |sentences|$ and $os \neq$ null then
            $i \leftarrow i - 1$
        end if
    end if
end while
return $\mathcal{C}$
```

and accuracy in summary generation. So, an open source punctuation model available on *Kaggle* (Tugstugi, 2023) was applied to insert punctuations via the *BanglaPunctuation* class from the *banglanlptoolkit* package.

3.3 Summary and Title Generation

We fine-tuned four Bengali summarization models and selected bT5 for its superior performance. Based on Google's T5 architecture, bT5 is specifically adapted for Bengali language processing (Hayat et al., 2023). For title generation, the *mT5-multilingual-XLSum* model performed best. It is based on Google's multilingual T5 framework fine-tuned for cross-lingual summarization across 101 languages, including Bengali (Xue et al., 2021).

For summarization, the bT5 model's 512-token limit required splitting long video transcriptions into manageable chunks. So, the punctuated transcription was split into sentence-aware token chunks with one-sentence overlap (Algorithm 2). The overlapping method was chosen over non-overlap chunking to preserve context across boundaries. In progressive summarization, each chunk is summarized, then its summary is combined with the next chunk and summarized again, repeating the process until the final summary is obtained. The overlapping strategy avoids this repeated processing, thereby reducing computational overhead. The metrics summarized in Table 2 and in A.2.1, also indicate that the overlapping strategy outperforms the others.

Metrics	Overlap	Non-overlap	Progressive
ROUGE-1 (F1)	**0.5360**	0.4930	0.3480
ROUGE-2 (F1)	**0.2031**	0.1703	0.1069
ROUGE-L (F1)	**0.2055**	0.1998	0.1720
BERTScore (F1)	0.8759	**0.8765**	0.8625
MoverScore	**0.4980**	**0.4980**	0.4845

Table 2: Comparison between 3 Chunking Techniques. Bold values represent the best values.

Normalization Type	Bengali Example	Converted Form
Number	(১, এক),(৫, ফাইভ)	1, 5
Mathematical notation	সাইন থিটা, প্লাস, স্কয়ার	sin θ, +, ²
Chemical formula	হাইড্রোক্লোরিক এসিড	HCl
	সোডিয়াম ক্লোরাইড	NaCl
Alphabet and variables	বি, থিটা, পাই	b, θ, π
Unit standardization	সেকেন্ড, কেজি	sec, kg

Table 3: Examples of Regex for Postprocessing.

The smart chunks were then fed into the fine-tuned bT5 model for abstractive summary generation using beam search (4 beams) with output lengths constrained between 100–200 tokens, a length penalty of 1.0, and n-gram repetition limited to 3, with early stopping and deterministic decoding (*do_sample = False*). Inference is run under *torch.no_grad()* for efficiency, and outputs are decoded to text with special tokens removed.

For long videos with more than 50 text chunks, recursive summarization was used to maintain coherence and efficiency (Figure 3). Each chunk is

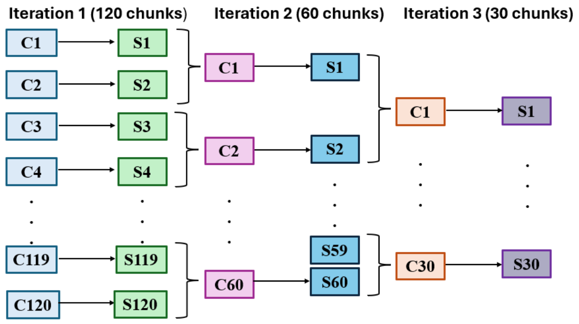


Figure 3: Recursive Summarization. C and S denote chunk and summary, respectively.

first summarized with bT5 model, then adjacent chunks are merged and re-summarized until the count drops below 50.

The bT5 model was fine-tuned for 10 epochs using the AdamW optimizer with a learning rate of 2e-5, batch size of 4 for both training and evaluation, and weight decay of 0.01. All experiments were conducted on a single NVIDIA T4 GPU using Google Colab.

For title generation, the generated summary is tokenized and truncated to a maximum of 1024 tokens to meet the model's processing constraints, and then fed into the fine-tuned *mT5-multilingual-XLSum* model. Beam search decoding was employed to explore multiple candidate sequences in parallel (Equation 1):

$$\text{Title} = \arg\max_{Y} \sum_{t=1}^{T} \log P(y_t \mid y_{<t}, X_{\text{sum}}) \quad (1)$$

Here, sum denotes summary, beam size is 4, titles are constrained to a minimum of 5 tokens and a maximum of 30 tokens, the default value of length Penalty is 1.0, and early stopping is enabled to halt decoding once all beams have generated an end-of-sequence token.

3.4 Data Post-processings

$$t^{start}_{sentence} = t^{start}_{chunk} + \left(\frac{i_{sentence}}{n_{sentences}} \times d_{chunk} \right) \quad (2)$$

$$t^{end}_{sentence} = t^{start}_{chunk} + \left(\frac{i_{sentence} + 1}{n_{sentences}} \times d_{chunk} \right) \quad (3)$$

Model	WER	CER
Wav2vec2-large-xlsr 53	1.1751	1.9120
Facebook-mms-102	0.5876	0.8820
Facebook-mms-All	0.5537	0.9390
Huggingface-whisper-large-v3	1.3446	1.6920
Openai-whisper-large-v3 (package)	1.3669	1.9080
Whisper-small- Bangla	1.0678	1.6200
Google ASR	0.3672	0.3380
Google ASR (pre-processed)	**0.2825**	**0.3378**

Table 4: Transcription Models Comparison.

Metric		bT5	NLLB	mBART	mT5
R-1(F1)	Mean ± HW	**0.4018** ± 0.0057	0.3629 ± 0.0056	0.3438 ± 0.0047	0.3400 ± 0.0053
R-2(F1)	Mean ± HW	**0.1645** ± 0.0046	0.1187 ± 0.0040	0.1044 ± 0.0034	0.1139 ± 0.0037
R-L(F1)	Mean ± HW	**0.2579** ± 0.0043	0.2249 ± 0.0040	0.2016 ± 0.0036	0.2015 ± 0.0034
BERT(F1)	Mean ± HW	**0.8795** ± 0.0009	0.8734 ± 0.0009	0.8686 ± 0.0010	0.8694 ± 0.0008
Mover(avg)	Mean ± HW	**0.6036** ± 0.0047	0.5899 ± 0.0045	0.5812 ± 0.0038	0.5830 ± 0.0043
F. C.	Factcc	0.2064	0.2356	**0.2442**	0.2043
	Dae	**0.9409**	0.9278	0.9245	0.9343
	SummaC	**0.7429**	0.7122	0.7078	0.7152

Table 5: Fine-tuned Summarization Models Comparison. HW denotes Half Width.

Integrating timestamps into summarized segments enables precise alignment with the original video for efficient navigation (Algorithm 3 in the appendix). Audio is divided into 45-second chunks, each assigned millisecond-level start and end times to preserve chronological flow without gaps or overlaps. Sentence-level timestamps within each chunk are distributed proportionally (Equations 2 and 3), and carry-over text across chunks is managed by dynamically adjusting temporal boundaries.

The final summary and title are fully in Bengali. To handle formulas, symbols, and mathematical notations typically in English, regex-based conversions standardize them for readability, as summarized in Table 3. The complete academic video summarization pipeline is shown in Algorithm 4 and Figure 6 (Appendix).

4 Main Results

For transcription, we use Word Error Rate (WER) and Character Error Rate (CER). Summary evaluation employs ROUGE, BERTScore, MoverScore, and factual consistency to compare *BanglaT5* (bT5), *mBART-50* (mBART), *NLLB 200-Distilled* (NLLB), and *mT5 (small)* (mT5). ROUGE measures lexical overlap, BERTScore and MoverScore assess semantic similarity, and factual consistency evaluates preservation of source facts. Title evaluation uses ROUGE for bT5, mBART, mT5, and

Models	R-1(F1)	R-2(F1)	R-L(F1)	BERT(F1)	Mover
bT5	**0.5275**	**0.2025**	**0.2211**	**0.883814**	**0.5649**
NLLB	0.4514	0.1314	0.1790	0.874692	0.5143
mBART	0.4620	0.1333	0.1683	0.87339	0.5472
mT5	0.4361	0.1434	0.1789	0.8655	0.4736
LLaVa	0.0058	0.0000	0.0058	0.7887	0.3706
Qwen	0.0116	0.0000	0.0103	0.8102	0.3606
Phi	0.0016	0.0000	0.0016	0.8022	0.3304
Llama	0.2770	0.0371	0.0932	0.8503	0.4472

Table 6: Results for Full Video Summarization.

Metric	Human Summary	Generated Summary
Grammatical Consistency	2.8729	2.2271
Information Retrieval	2.6971	2.3029
Factual Consistency	2.7814	2.4571
Con	2.5814	2.4014

Table 7: Likert Scores for Summary Evaluation.

Metric		BANS	benSumm	Ours
R-1	Precision	0.0734	0.2560	**0.4668**
	Recall	0.0577	0.1618	**0.3430**
	F1 score	0.0630	0.1904	**0.3894**
R-2	Precision	0.0150	0.0707	**0.1935**
	Recall	0.0120	0.0436	**0.1417**
	F1 score	0.0130	0.0516	**0.1610**
R-L	Precision	0.0576	0.1657	**0.3062**
	Recall	0.0455	0.1024	**0.2254**
	F1 score	0.0496	0.1215	**0.2557**
BERT	Precision	0.7956	0.8487	**0.8854**
	Recall	0.8137	0.8331	**0.8735**
	F1 score	0.8508	0.8407	**0.8793**
Mover	Average	0.3924	0.4904	**0.5731**

Table 8: Evaluation for Different Datasets

mT5-multilingual-XLSum (mT5-XL). Details of the metrics are in Appendix A.3. Bold table values indicate the best-performing configurations. For brevity, in all tables in this section, ROUGE scores, BERTScores, MoverScore, and factual consistency metrics are denoted as R, BERT, Mover, and F.C., respectively.

4.1 Evaluation of Transcription

Table 4 compares several ASR models based on WER and CER. Among them, the Google ASR system achieves the lowest error rates (WER: 0.3672, CER: 0.3380), while its pre-processed version performs best overall with WER: 0.2825 and CER: 0.3378. This indicates that audio pre-processing improves Google's ASR performance by approximately 23% in WER. The qualitative result is tabulated in Appendix Table 21. Further analysis on the effect of ASR choice on downstream generation tasks is provided in Appendix A.8.

Metric		*mBART*	*bT5*	*mT5*	*mT5-XL*
R-1	Precision	0.0152	0.0766	0.0170	**0.0792**
	Recall	0.0057	**0.1250**	0.0076	0.0649
	F1 score	0.0083	**0.0938**	0.0104	0.0700
R-2	Precision	0.0000	0.0052	0.0000	**0.0792**
	Recall	0.0000	0.0076	0.0000	**0.0649**
	F1 score	0.0000	0.0062	0.0000	**0.0700**
R-L	Precision	0.0000	0.0052	0.0000	**0.0792**
	Recall	0.0000	0.0076	0.0000	**0.0649**
	F1 score	0.0000	0.0062	0.0000	**0.0700**

Table 9: Raw Model Comparison for Title.

Metric		*bT5*	*mT5-XL*
R-1	Precision	0.0867	**0.5978**
	Recall	0.1327	**0.3711**
	F1 score	0.1033	**0.4476**
R-2	Precision	0.0120	**0.2966**
	Recall	0.0159	**0.1734**
	F1 score	0.0136	**0.2129**
R-L	Precision	0.0786	**0.4886**
	Recall	0.1223	**0.3110**
	F1 score	0.0942	**0.3720**

Table 10: Fine-Tuned Model Comparison for Title.

4.2 Evaluation of Text Summarization

Table 5 compares four fine-tuned Bengali summarization models. HW (Half Width) represents half of the 95% Confidence Interval, estimated via the *bootstrap* method; smaller HW values indicate more reliable scores. Overall, bT5 performs best, achieving the highest scores (e.g., ROUGE-1= 0.4018, BERTScore= 0.8795), and strong F.C.. NLLB ranks second, showing competitive scores (e.g., ROUGE-1 = 0.3629, BERTScore = 0.8734), while mBART performs worst across most metrics despite slightly higher FactCC (0.2442). bT5 outperforms the second-best NLLB by approximately 10.7% in ROUGE-1, 38.6% in ROUGE-2, and 14.7% in ROUGE-L.

In Table 6, we present an evaluation of full-video summaries generated using the four previously fine-tuned models as well as four recent models: *LLaVa v1.6 mistral* (LLaVa), *Qwen 2.5-1.5B instruct* (Qwen), *Phi 3- mini 128K instruct* (phi), and *Llama 7B instruct v01* (Llama). The recent models are evaluated in their raw form, as fine-tuning them requires huge computational resources and large-scale datasets. Table 6 shows that fine-tuned Bengali models vastly outperform recent raw models on full-video summarization, with bT5 leading (ROUGE-1: 0.528, BERTScore: 0.884) and Llama the only recent model showing moderate performance (ROUGE-1: 0.277, BERTScore: 0.850). LLaVa, Qwen, and Phi score near zero on ROUGE. It indicates the need for fine-tuning on

Metric		bT5(r)	bT5(ft)	bT5(Im)	mB(r)	mB(ft)	mB(Im)	NL(r)	NL(ft)	NL(Im)	mT5(r)	mT5(ft)	mT5(Im)
R-1	Pre	0.1314	**0.4295**	226.8%	0.2478	**0.3488**	40.8%	0.0007	**0.3992**	57,000%	0.1382	**0.3847**	178.3%
	Rec	0.0782	**0.3871**	395.0%	0.0859	**0.3494**	306.9%	0.0005	**0.3426**	68,420%	0.0435	**0.3128**	619.1%
	F1	0.0963	**0.4018**	317.2%	0.1202	**0.3438**	186.0%	0.0006	**0.3629**	60,383%	0.0652	**0.3400**	421.5%
R-2	Pre	0.0294	**0.1756**	497.3%	0.0709	**0.1063**	49.9%	0.0001	**0.1305**	130,400%	0.0247	**0.1293**	423.5%
	Rec	0.0174	**0.1587**	812.1%	0.0219	**0.1056**	382.2%	0.0001	**0.1120**	111,900%	0.0076	**0.1044**	1,273.7%
	F1	0.0215	**0.1645**	665.1%	0.0312	**0.1044**	234.6%	0.0002	**0.1187**	59,250%	0.0114	**0.1139**	899.1%
R-L	Pre	0.1013	**0.2750**	171.5%	0.1868	**0.2047**	9.6%	0.0007	**0.2473**	35,229%	0.1105	**0.2277**	106.1%
	Rec	0.0602	**0.2491**	313.8%	0.0614	**0.2048**	233.6%	0.0005	**0.2124**	42,380%	0.0348	**0.1857**	433.6%
	F1	0.0742	**0.2579**	247.6%	0.0869	**0.2016**	132.0%	0.0006	**0.2249**	37,383%	0.0522	**0.2015**	286.0%
BS	Pre	0.8178	**0.8827**	7.9%	0.8178	**0.8827**	7.9%	0.8153	**0.8766**	7.5%	0.8240	**0.8745**	6.1%
	Rec	0.8206	**0.8765**	6.8%	0.8206	**0.8765**	6.8%	0.8185	**0.8703**	6.3%	0.8038	**0.8645**	7.6%
	F1	0.8191	**0.8795**	7.4%	0.8191	**0.8795**	7.4%	0.8167	**0.8734**	6.9%	0.8136	**0.8694**	6.9%
MS	Avg	0.3916	**0.6036**	54.2%	0.4823	**0.5812**	20.5%	0.1517	**0.5899**	288.9%	0.3924	**0.5830**	48.6%

Table 11: Raw vs Fine-tuned Models Comparisons with Improvement Percentages for Summarization. BS, MS, mB, and NL denote BERTScore, MoverScore, mBART, and NLLB, respectively. Bold values represent the best values achieved after fine-tuning each model.

domain-specific data.

Table 7 presents the average 3-point *Likert scale* (1=poor, 2=fair, 3=good) for human-written and model-generated summaries across grammatical consistency, information retrieval, factual consistency, and conciseness. The evaluation was conducted by seven independent reviewers who were blind to whether the summary was human or machine-generated. Table 7 reveals that the model can produce summaries that are largely fluent, informative, and factually consistent, as the model-generated summaries achieve scores close to human performance. The average scores of each reviewer for the 100 samples are shown in Table 19 (Appendix).

To evaluate whether existing formal text summarization datasets could adapt bT5 for academic video content, we fine-tuned the model on BANS (Bhattacharjee et al., 2020) and benSumm (Hasib et al., 2023) and compared results on our dataset (Table 8). The results show that formal text datasets perform poorly on spoken academic videos: ROUGE-1/2/L F1 scores are 0.063/0.013/0.050 (BANS) and 0.190/0.052/0.122 (benSumm) versus 0.389/0.161/0.256 for our dataset. BERTScore F1 rises from 0.851/0.841 to 0.879, and MoverScore from 0.392/0.490 to 0.573, highlighting the need for a domain-specific dataset. The qualitative result for summarization is shown in Appendix A.9.2.

4.3 Evaluation of Title Generation

We evaluated four pre-trained models in Table 9 to assess their baseline Bengali academic title generation performance without domain-specific fine-tuning. Among them, mT5-XL achieves the highest scores across all ROUGE metrics, with ROUGE-1 F1 at 0.0700, ROUGE-2 F1 at 0.0700, and ROUGE-L F1 at 0.0700, showing superior precision and recall compared to bT5 (ROUGE-1 F1: 0.0938, ROUGE-2 F1: 0.0062, ROUGE-L F1: 0.0062) and the other models. mBART and mT5 perform poorly, with near-zero ROUGE-2 and ROUGE-L scores.

Based on the raw model evaluation, *bT5* and *mT5-XL* were identified as the top-performing models. After fine-tuning on our benchmark dataset, mT5-XL clearly outperforms bT5 on the test set, achieving ROUGE-1 F1 of 0.448 versus 0.1033, ROUGE-2 F1 of 0.213 versus 0.0136, and ROUGE-L F1 of 0.372 versus 0.0942 (Table 10,18). The qualitative result for title generation is shown in Appendix A.9.3, and an example of final output is shown in Appendix A.9.4.

4.4 Ablation Study

This section provides valuable insights into error analysis, inference time comparison, and zero-shot capabilities of the fine-tuned bT5 model.

4.4.1 Error Analysis

Table 11 presents a comparison of raw(r) and fine-tuned(ft) model performances, with percentage improvement(Im) highlighting the gains from fine-tuning. For our proposed bT5 model, fine-tuning leads to improvements across all metrics, particularly in ROUGE scores (ROUGE-1 F1: 0.0963 → 0.4018, 317.2%; ROUGE-2 F1: 0.0215 → 0.1645,

Metric		bT5(r)	bT5(ft)	bT5(Im)	mT5(r)	mT5(ft)	mT5(Im)
R-1	Pre	0.0766	**0.0867**	13.19%	0.0792	**0.5978**	654.80%
	Rec	0.1250	**0.1327**	6.16%	0.0649	**0.3711**	471.80%
	F1	0.0938	**0.1033**	10.13%	0.0700	**0.4476**	539.43%
R-2	Pre	0.0052	**0.0120**	130.77%	0.0792	**0.2966**	274.49%
	Rec	0.0076	**0.0159**	109.21%	0.0649	**0.1734**	167.18%
	F1	0.0062	**0.0136**	119.35%	0.0700	**0.2129**	204.14%
R-L	Pre	0.0052	**0.0786**	1411.54%	0.0792	**0.4886**	516.92%
	Rec	0.0076	**0.1223**	1509.21%	0.0649	**0.3110**	379.20%
	F1	0.0062	**0.0942**	1419.35%	0.0700	**0.3720**	431.43%

Table 12: Raw vs Fine-tuned Models Comparisons with Improvement Percentages for Title Generation.

Content Genre	ROUGE-L(F1)	BERTScore(F1)	MoverScore
News Content	0.1743	0.8974	0.5505
Podcasts	0.1353	0.8814	0.5505
Tech-Related Videos	0.1632	0.8837	0.5318
Other Academic Videos	**0.2095**	**0.8712**	**0.6089**

Table 13: Cross-domain Performance for Summarization.

665.1%). This reflects a significant reduction in lexical errors. Improvements in BERTScore and MoverScore (F1: 0.8191 → 0.8795, 7.4%; 0.3916 → 0.6036, 54.2%) reflect enhanced semantic consistency. NLLB and other models show enormous percentage improvements due to extremely low raw scores; bT5 achieves the highest absolute performance as its smaller size allows effective domain-specific fine-tuning on our 10,029 text-summary pairs.

Similarly, for title generation, Table 12 reveals that bT5 has modest gains (ROUGE-1 F1: 10.13%), but mT5-XL (here denoted as mT5) achieves the highest absolute performance after fine-tuning with very large improvements (ROUGE-1 F1: 539.43%). In both Table 11, and 12, Pre, Rec, and F1 denote Precision, Recall, and F1 Score, respectively.

4.4.2 Inference Time Analysis

Figure 4 compares model inference times. bT5 is fastest at 2.54 seconds. It is computationally efficient due to its fewer parameters compared to the other models (Appendix Table 17). Qwen is the slowest at 78.32 seconds, over 30 times longer.

We analyze the end-to-end inference latency of the proposed video summarization pipeline on approximately one-hour videos (Table 14). The audio-to-text (ASR) stage dominates the overall runtime, while text summarization and title generation incur comparatively low latency. On average, the full pipeline completes in under 14 minutes per video.

Pipeline Stage	Mean (s)	Std (s)
Video Preprocessing	69.0	39.2
Video → Audio	31.3	1.2
Audio → Text (ASR)	631.3	36.6
Text → Summary	51.3	3.8
Title Generation	45.7	3.1
Total Inference Time	**828.7**	**55.6**

Table 14: Mean and standard deviation (Std) of inference time (in seconds) for each stage of the end-to-end video summarization pipeline, evaluated on approximately one-hour videos.

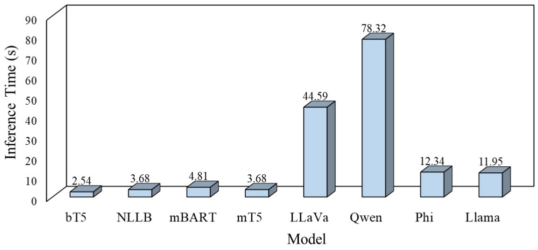


Figure 4: Inference Time Comparison for Summarization. bT5, mBART, NLLB, and mT5 represent fine-tuned models, while the others are used without fine-tuning.

4.4.3 Cross-domain Performance Analysis for Summarization

Table 13 presents the zero-shot summarization performance of the fine-tuned bT5 model across four content genres. The model achieves the highest ROUGE-L F1 (0.2095), BERTScore (0.8712), and MoverScore (0.6089) on other academic videos outside the training domain. It reflects a strong generalization of it to unseen academic content.

5 Conclusion

This study presents the first comprehensive Bengali academic video summarization system, combining fine-tuned *BanglaT5* and *mT5-multilingual-XLSum* models for abstractive summarization and title generation using our benchmark datasets. While transcription accuracy and real-time processing remain challenging, the pipeline provides a pipeline for Bengali video summarization, contributes valuable resources to low-resource NLP, and enhances accessibility to educational content.

Limitations

This study establishes a complete pipeline for Bengali academic video summarization, marking a novel contribution to research in this domain. While the proposed system achieves promising results and demonstrates the feasibility of the task, several limitations remain that open up directions for further improvement.

In this work, we primarily focused on text-based summarization derived from video subtitles rather than generating summaries in the form of video content. Future research can explore multimodal summarization that combines visual, audio, and textual features to produce short, meaningful video clips. Although we have included videos from different speakers to address dialectal and subtitle-related biases, the diversity of Bengali dialects still poses challenges that require broader representation in future datasets. Expanding the dataset with more inclusive linguistic and contextual diversity would make the system more accessible to a wider audience. Additionally, future advancements can focus on improving the speech-to-text component to minimize transcription errors, which often propagate into the summaries. Lastly, real-time summarization for live sessions presents another promising direction. It requires efficient optimization and computational support to process continuous input streams with minimal latency.

Ethics Statement

All videos used in this study were collected from publicly available YouTube playlists with the content owners' consent and in accordance with YouTube's data usage policies. The dataset was curated solely for academic research and does not include any personally identifiable or sensitive information. AI generative tools, such as OpenAI's ChatGPT and Grammarly, were used solely to enhance the clarity and grammar of the manuscript. No text or data were generated or altered in a manner that affects the scientific validity or originality of the research.

References

Marah I Abdin, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, Hany Hassan Awadalla, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Harkirat Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Martin Cai, Caio César Teodoro Mendes, Weizhu Chen, Vishrav Chaudhary, Parul Chopra, Allie Del Giorno, Gustavo de Rosa, Matthew Dixon, Ronen Eldan, Dan Iter, Abhishek Goswami, Suriya Gunasekar, Emman Haider, Junheng Hao, Russell J. Hewett, Jamie Huynh, Mojan Javaheripi, Xin Jin, Piero Kauffmann, Nikos Karampatziakis, Dongwoo Kim, Mahmoud Khademi, Lev Kurilenko, James R. Lee, Yin Tat Lee, Yuanzhi Li, Chen Liang, Weishung Liu, Xihui (Eric) Lin, Zeqi Lin, Piyush Madan, Arindam Mitra, Hardik Modi, Anh Nguyen, Brandon Norick, Barun Patra, Daniel Perez-Becker, Thomas Portet, Reid Pryzant, Heyang Qin, Marko Radmilac, Corby Rosset, Sambudha Roy, Olli Saarikivi, Amin Saied, Adil Salim, Michael Santacroce, Shital Shah, Ning Shang, Hiteshi Sharma, Xia Song, Olatunji Ruwase, Xin Wang, Rachel Ward, Guanhua Wang, Philipp Witte, Michael Wyatt, Can Xu, Jiahang Xu, Weijian Xu, Sonali Yadav, Fan Yang, Ziyi Yang, Donghan Yu, Chengruidong Zhang, Cyril Zhang, Jianwen Zhang, Li Lyna Zhang, Yi Zhang, Yunan Zhang, and Xiren Zhou. 2024. Phi-3 technical report: A highly capable language model locally on your phone. Technical report, Microsoft.

Abdullah Al Maruf, Ahmad Jainul Abidin, Md Mahmudul Haque, Zakaria Masud Jiyad, Aditi Golder, Raaid Alubady, and Zeyar Aung. 2024. Hate speech detection in the bengali language: a comprehensive survey. *Journal of Big Data*, 11(1):97.

Sarah S Alrumiah and Amal A Al-Shargabi. 2022. Educational videos subtitles' summarization using latent dirichlet allocation and length enhancement. *Computers, Materials & Continua*, 70(3).

Evlampios Apostolidis, Eleni Adamantidou, Alexandros I Metsai, Vasileios Mezaris, and Ioannis Patras. 2021a. Video summarization using deep neural networks: A survey. *Proceedings of the IEEE*, 109(11):1838–1863.

Evlampios Apostolidis, Eleni Adamantidou, Alexandros I Metsai, Vasileios Mezaris, and Ioannis Patras. 2021b. Video summarization using deep neural networks: A survey. *Proceedings of the IEEE*, 109(11):1838–1863.

Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Yu Bai, Yang Gao, and Heyan Huang. 2021. Cross-lingual abstractive summarization with limited parallel resources. *arXiv preprint arXiv:2105.13648*.

Marcello Barbella and Genoveffa Tortora. 2022. Rouge metric evaluation for text summarization techniques. *Available at SSRN 4120317*.

Shashank Bhargav, Abhinav Choudhury, Shruti Kaushik, Ravindra Shukla, and Varun Dutt. 2022. A comparison study of abstractive and extractive methods for text summarization. In *Proceedings* *of the International Conference on Paradigms of Communication, Computing and Data Sciences: PCCDS 2021*, pages 601–610. Springer.

Abhik Bhattacharjee, Tahmid Hasan, Wasi Uddin Ahmad, and Rifat Shahriyar. 2023. BanglaNLG and BanglaT5: Benchmarks and resources for evaluating low-resource natural language generation in Bangla. In *Findings of the Association for Computational Linguistics: EACL 2023*, Dubrovnik, Croatia. Association for Computational Linguistics.

Prithwiraj Bhattacharjee, Avi Mallick, and Md Saiful Islam. 2020. Bengali abstractive news summarization (bans): a neural attention approach. In *Proceedings of International Conference on Trends in Computational and Cognitive Engineering: Proceedings of TCCE 2020*, pages 41–51. Springer.

Alexey Bukhtiyarov and Ilya Gusev. 2020. Advances of transformer-based models for news headline generation. In *Conference on Artificial Intelligence and Natural Language*, pages 54–61. Springer.

Radia Rayan Chowdhury, Mir Tafseer Nayeem, Tahsin Tasnim Mim, Md. Saifur Rahman Chowdhury, and Taufiqul Jannat. 2021. Unsupervised abstractive summarization of Bengali text documents. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, Online. Association for Computational Linguistics.

Ensieh Davoodijam and Mohsen Alambardar Meybodi. 2024. Evaluation metrics on text summarization: comprehensive survey. 66(12).

Daniel Deutsch and Dan Roth. 2021. Understanding the extent to which content quality metrics measure the information quality of summaries. In *Proceedings of the 25th Conference on Computational Natural Language Learning*, Online. Association for Computational Linguistics.

Prakash Dhakal and Daya Sagar Baral. 2024. Abstractive summarization of low resourced nepali language using multilingual transformers. *arXiv preprint arXiv:2409.19566*.

Hana Gharbi, Sahbi Bahroun, and Ezzeddine Zagrouba. 2019. Key frame extraction for video summarization using local description and repeatability graph clustering. *Signal, Image and Video Processing*, 13(3):507–515.

Nikolaos Giarelis, Charalampos Mastrokostas, and Nikos Karacapilidis. 2023. Abstractive vs. extractive summarization: An experimental review. *Applied Sciences*, 13(13):7620.

Tahmid Hasan, Abhik Bhattacharjee, Md Saiful Islam, Kazi Samin, Yuan-Fang Li, Yong-Bin Kang, M Sohel Rahman, and Rifat Shahriyar. 2021. Xl-sum: Large-scale multilingual abstractive summarization for 44 languages. *arXiv preprint arXiv:2106.13822*.

Khan Md Hasib, Md. Atiqur Rahman, Mustavi Ibne Masum, Friso De Boer, Sami Azam, and Asif Karim. 2023. Bengali news abstractive summarization: T5 transformer and hybrid approach. In *2023 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, pages 539–545.

S. M. Afif Ibne Hayat, Avishek Das, and Mohammed Moshiul Hoque. 2023. Abstractive bengali text summarization using transformer-based learning. In *2023 6th International Conference on Electrical Information and Communication Technology (EICT)*, pages 1–6.

Rosalind Horowitz and S Jay Samuels. 2023. *Comprehending oral and written language*. Brill.

Hyun Hee Kim and Yong Ho Kim. 2016. Generic speech summarization of transcribed lecture videos: Using tags and their semantic relations. *Journal of the Association for Information Science and Technology*, 67(2):366–379.

Lakshmi Prasanna Kumar and Arman Kabiri. 2022. Meeting summarization: A survey of the state of the art. *arXiv preprint arXiv:2212.08206*.

Jinpeng Li, Jiaze Chen, Huadong Chen, Dongyan Zhao, and Rui Yan. 2024a. Multilingual generation in abstractive summarization: A comparative study. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 11827–11837, Torino, Italia. ELRA and ICCL.

Jinpeng Li, Jiaze Chen, Huadong Chen, Dongyan Zhao, and Rui Yan. 2024b. Multilingual generation in abstractive summarization: A comparative study. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 11827–11837.

LOKESH MADASU. 2024. *Headline Generation for Indian Languages*. Ph.D. thesis, International Institute of Information Technology Hyderabad.

Larisa Mamedova, Alexander Rukovich, Tetiana Likhouzova, and Lubov Vorona-Slivinskaya. 2023. Online education of engineering students: Educational platforms and their influence on the level of academic performance. *Education and Information Technologies*, 28(11).

Salman Masih, Mehdi Hassan, Labiba Gillani Fahad, and Bilal Hassan. 2025. Transformer-based abstractive summarization of legal texts in low-resource languages. *Electronics*, 14(12):2320.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2022. Robust speech recognition via large-scale weak supervision.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2023. Exploring the limits of transfer learning with a unified text-to-text transformer.

Sebastian Raschka. 2020. Model evaluation, model selection, and algorithm selection in machine learning.

Anuj Razdan, Praveen Kumar, Shaveta Bhatia, Nripendra Narayan Das, Alibek Orynbek, and Mohamed Ibrahim. 2024. Audio-enhanced video-to-audio retrieval using text-conditioned feature alignment. In *2024 International Conference on Computing, Sciences and Communications (ICCSC)*, pages 1–5. IEEE.

Rajeev Kumar Singh, Sonia Khetarpaul, Rohan Gorantla, and Sai Giridhar Allada. 2021. Sheg: summarization and headline generation of news articles using deep learning. *Neural Computing and Applications*, 33(8):3251–3265.

Md Talukder, Sheikh Abujar, Abu Kaisar Mohammad Masum, and Syed Hossain. 2019. Bengali abstractive text summarization using sequence to sequence rnns.

L. Tang, Z. Sun, B. Idnay, J. G. Nestor, A. Soroush, P. A. Elias, Z. Xu, Y. Ding, G. Durrett, J. Rousseau, C. Weng, and Y. Peng. 2023. Evaluating large language models on medical evidence summarization. *medRxiv*, page 2023.04.22.23288967.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models.

Tugstugi. 2023. bengali-ai-asr-submission. https://www.kaggle.com/datasets/tugstugi/bengali-ai-asr-submission/data. Kaggle dataset, Accessed: 2025-08-24.

Florența-Diana Tănase, Suzana Demyen, Venera-Cristina Manciu, and Adrian-Costinel Tănase. 2022. Online education in the covid-19 pandemic—premise for economic competitiveness growth? *Sustainability*, 14(6).

Pascal Wilman, Talia Atara, and Derwin Suhartono. 2024. Abstractive english document summarization using bart model with chunk method. *Procedia Computer Science*, 245:1010–1019.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Online. Association for Computational Linguistics.

Jun Yang, Shifan Liu, Qifan He, Songcheng Xie, and Zhanqi Cui. 2024. Issue title generation: How far can large language models go? In *2024 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 2480–2485. IEEE.

Fangli Ying, Ziyue Luo, and Aniwat Phaphuangwittayakul. 2024. Enhancing multimodal video summarization via temporal and semantic alignment. In *International Conference on Neural Information Processing*, pages 17–31. Springer.

Haopeng Zhang, Philip S. Yu, and Jiawei Zhang. 2025. A systematic survey of text summarization: From statistical methods to large language models. 57(11).

Mengli Zhang, Gang Zhou, Wanting Yu, Ningbo Huang, and Wenfen Liu. 2022. A comprehensive survey of abstractive text summarization based on deep learning. *Computational intelligence and neuroscience*, 2022(1):7132226.

Ruqing Zhang, Jiafeng Guo, Yixing Fan, Yanyan Lan, and Xueqi Cheng. 2020. Structure learning for headline generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9555–9562.

Tianyi Zhang, Faisal Ladhak, Esin Durmus, Percy Liang, Kathleen McKeown, and Tatsunori B. Hashimoto. 2024. Benchmarking large language models for news summarization. *Transactions of the Association for Computational Linguistics*, 12.

Yu Zhang, Wei Qin, Daniel S. Park, Chung-Cheng Han, Chung-Cheng Chiu, Ruoming Pang, Quoc V. Le, and Yonghui Wu. 2023. Google USM: Scaling automatic speech recognition beyond 100 languages. *arXiv preprint arXiv:2303.01037*.

A Appendix

A.1 Related Work

A significant amount of research has been conducted on extractive and abstractive summarization. They employed classical and transformer-based architectures. Earlier works mostly used rule-based or statistical approaches, while recent advances rely on deep learning and pre-trained language models. The table 15 presents an overview of existing summarization models, their methods, and corresponding limitations. This is a concise form of section 2.

A.2 Techniques and Pipeline

Our paper uses audio pre-processing to improve transcription quality, along with timestamp integration for increasing readability. The complete process involves several steps to generate the final output. The Figure 5 represents the audio pre-processing technique discussed in section 3.2. The timestamp integration technique mentioned in section 3.4 is shown in detail using the Algorithm 3. The complete pipeline follows the Algorithm 4 and it is illustrated in Figure 6.

Algorithm 3 Timestamp Integration with Carry-over Management.

```
Require: Audio $A$, chunks $\mathcal{C} = \{c_1, \ldots, c_m\}$ with text
Ensure: Timestamped sentences $\mathcal{T}$
Initialize: $\mathcal{T} \leftarrow \emptyset$, $carry \leftarrow$ null, $t_c \leftarrow 0$, $id \leftarrow 1$
$T_{audio} \leftarrow$ Duration($A$), $\Delta_{chunk} \leftarrow 45000$ ms
Phase 1: Map Audio Chunks to Time
for $i = 1$ to $|\mathcal{C}|$ do
    $t_i^s \leftarrow (i-1) \cdot \Delta_{chunk}$, $t_i^e \leftarrow \min(i \cdot \Delta_{chunk}, T_{audio})$
    $x_i \leftarrow$ Transcribe($c_i$)
end for
Phase 2: Sentence-Level Timestamp Mapping
for $i = 1$ to $|\mathcal{C}|$ do
    $t_s \leftarrow t_i^s, t_e \leftarrow t_i^e, d \leftarrow t_e - t_s$
    if IsError($x_i$) then
        if $carry \neq$ null then
            $\mathcal{T} \leftarrow \mathcal{T} \cup \{(id, carry, t_c, t_s)\}$, $id \leftarrow id + 1$
        end if
        $\mathcal{T} \leftarrow \mathcal{T} \cup \{(id, x_i, t_s, t_e)\}$, $id \leftarrow id + 1$
        $carry \leftarrow$ null, continue
    end if
    $x_{full} \leftarrow$ Concat($carry, x_i$)
    $S \leftarrow$ Split($x_{full}$)                          ▷ By   ? !
    if $|S| = 0$ then
        $carry \leftarrow x_{full}$, $t_c \leftarrow (carry =$ null$)?t_s : t_c$
        continue
    end if
    $complete \leftarrow$ EndsPunct($x_{full}$) $\vee (i = |\mathcal{C}|)$
    if $complete$ then
        $S_{comp} \leftarrow S$, $carry_{new} \leftarrow$ null
    else
        $S_{comp} \leftarrow S[1 : |S| - 1]$, $carry_{new} \leftarrow S[|S|]$
    end if
    if $|S_{comp}| > 0$ then
        $t_{start} \leftarrow (carry \neq$ null$)?t_c : t_s$
        $D \leftarrow t_e - t_{start}$, $\delta \leftarrow D/|S_{comp}|$
        for $j = 1$ to $|S_{comp}|$ do
            $t_j^s \leftarrow t_{start} + (j-1) \cdot \delta$
            $t_j^e \leftarrow t_{start} + j \cdot \delta$
            $\mathcal{T} \leftarrow \mathcal{T} \cup \{(id, S_{comp}[j], t_j^s, t_j^e)\}$
            $id \leftarrow id + 1$
        end for
    end if
    $carry \leftarrow carry_{new}$
    if $carry_{new} \neq$ null then
        $t_c \leftarrow (|S_{comp}| > 0)?t_{start} + |S_{comp}| \cdot \delta : t_s$
    end if
end for
Finalize:
if $carry \neq$ null then
    $\mathcal{T} \leftarrow \mathcal{T} \cup \{(id, carry, t_c, t^e_{|\mathcal{C}|})\}$
end if
return $\mathcal{T}$
```

A.2.1 Overlap Length in Chunking

To minimize context fragmentation during summarization, our chunking module uses sentence-level overlaps. We initially adopted a one-sentence overlap based on the highly informal and segmented nature of the subtitle transcripts. To validate this choice, we conducted both qualitative and quantitative analyses comparing 1-, 2-, and 3-sentence overlaps. Table 16 reports redundancy and semantic similarity metrics across these configurations, showing that a one-sentence overlap introduced minimal repetition while preserving coherence. Overall, this setting provided the optimal trade-off between maintaining context continuity and avoiding excessive redundancy.

A.3 Evaluation Metrics

We have used multiple metrics to evaluate the quantitative results for summarizing and title generation. They inspect different aspects of the summaries and titles, for example, semantics, factual consistency, etc.

ROUGE Metrics: ROUGE (Recall-Oriented Understudy for Gisting Evaluation) represents the most widely adopted family of metrics for text summarization evaluation (Barbella and Tortora, 2022). These metrics measure the lexical overlap between generated summaries and reference summaries. It provides a standardized comparison across different summarization systems. Since traditional metrics like precision, recall, and F1 scores fail to evaluate summary generation properly, ROUGE is widely used for this task.

ROUGE-1: ROUGE-1 measures the overlap of unigrams (single words) between the generated

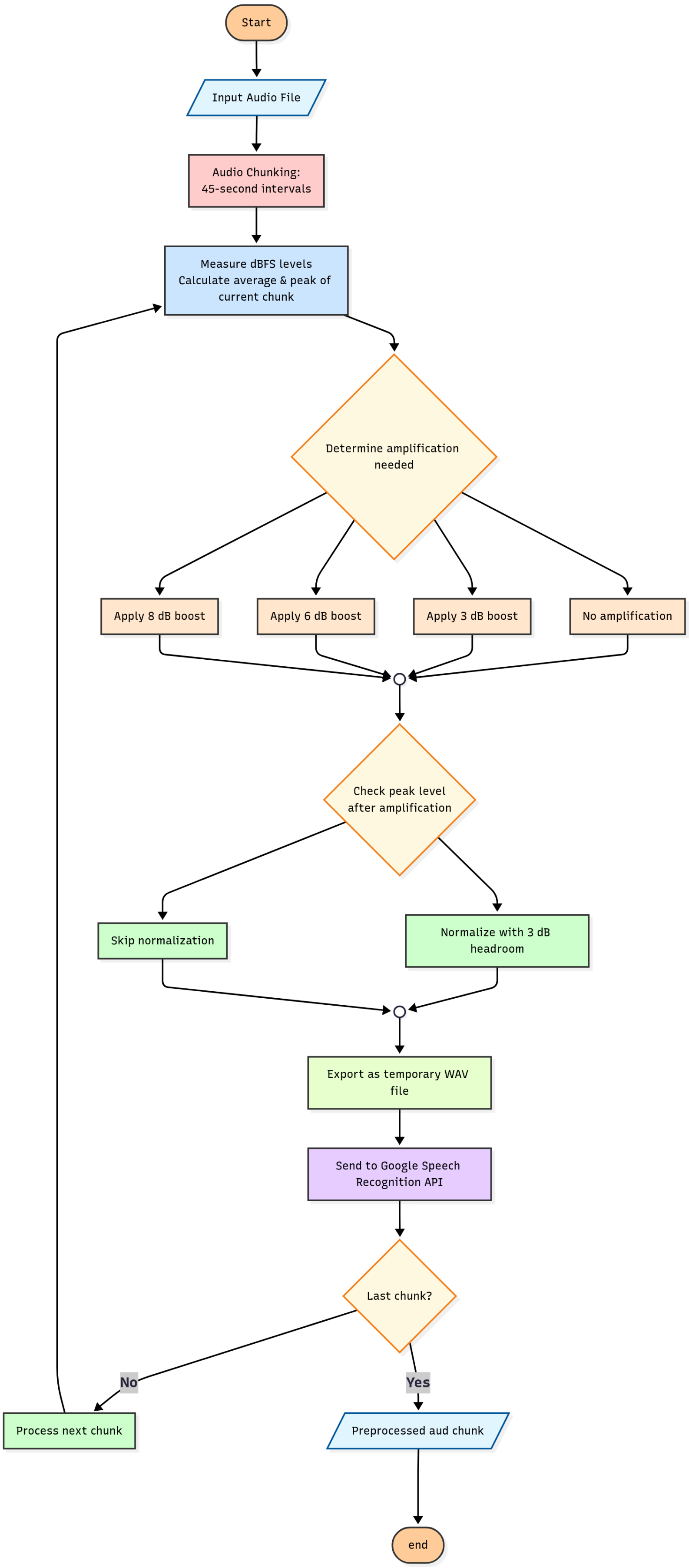


Figure 5: Audio Pre-processing Steps.

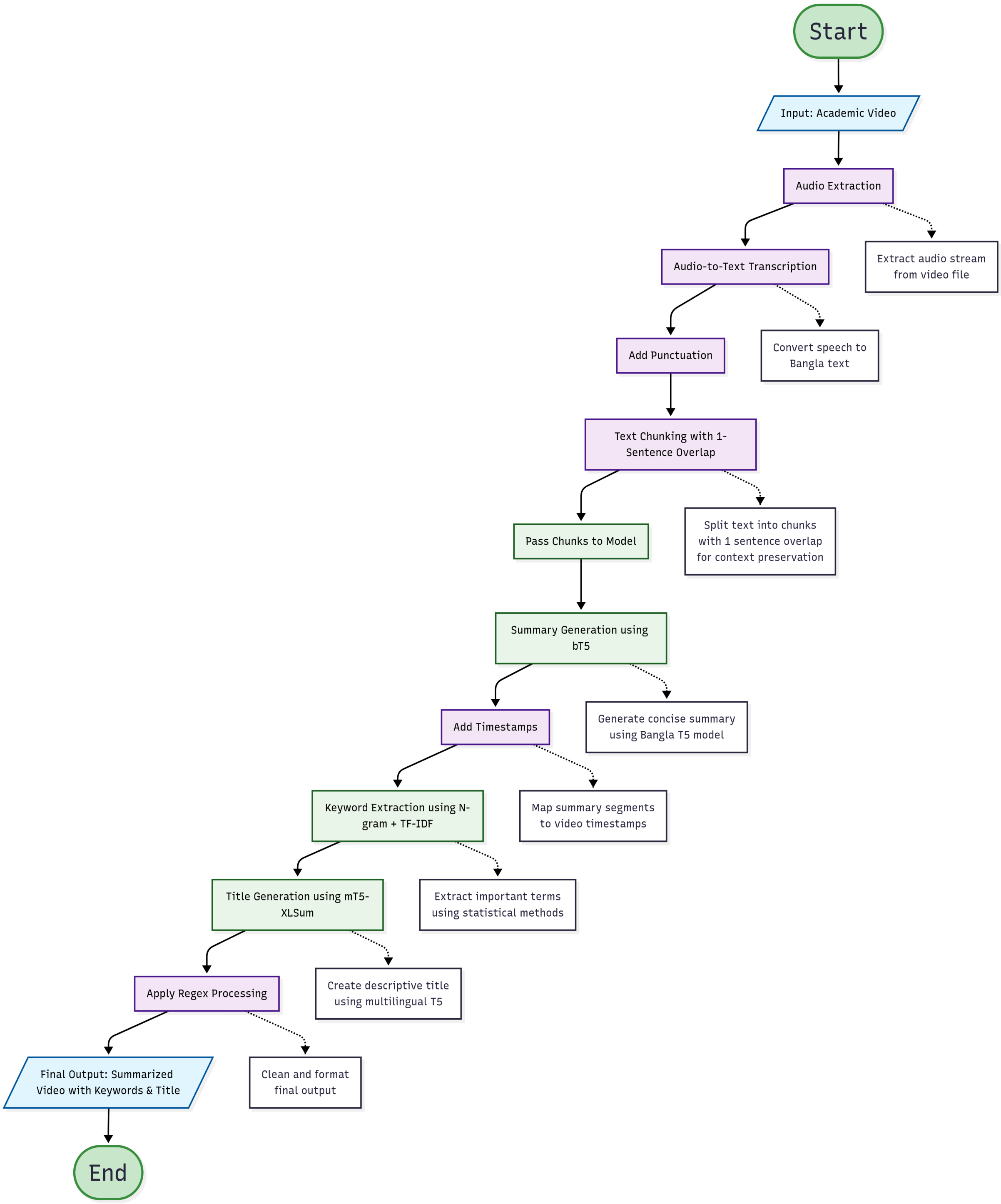

Start
Input: Academic Video
Audio Extraction
Extract audio stream from video file
Audio-to-Text Transcription
Convert speech to Bangla text
Add Punctuation
Text Chunking with 1-Sentence Overlap
Split text into chunks with 1 sentence overlap for context preservation
Pass Chunks to Model
Summary Generation using bT5
Generate concise summary using Bangla T5 model
Add Timestamps
Map summary segments to video timestamps
Keyword Extraction using N-gram + TF-IDF
Extract important terms using statistical methods
Title Generation using mT5-XLSum
Create descriptive title using multilingual T5
Apply Regex Processing
Clean and format final output
Final Output: Summarized Video with Keywords & Title
End

Figure 6: Pipeline for Summary Generation.

Authors	**Models**	**Weakness/Remarks**
Chowdhury et al.(Chowdhury et al., 2021)	Unsupervised BenSumm, clustering, word graph fusion	Better ROUGE in Bengali, but struggles with factual hallucinations and semantic drift.
Talukder et al. (Talukder et al., 2019)	Seq2seq, RNN, LSTM with attention	Requires larger datasets and optimization for longer texts.
Li et al.(Li et al., 2024a)	mBART for multilingual summarization	Struggles in low-resource languages, showing gaps in cross-lingual performance.
Xue et al.(Xue et al., 2021)	**mT5 for multilingual tasks (proposed for title generation)**	**Can generate coherent titles with less data due to crosslingual zero-shot support.**
Bhattacharjee et al.(Bhattacharjee et al., 2023)	***BanglaT5* (proposed for summary generation)**	**Outperforms other models in Bengali while containing a smaller number of parameters.**
Qwen Model Series (Bai et al., 2023)	QWEN and QWEN-CHAT for multilingual summarization	Strong in multilingual summarization, struggles with low-resource languages.
LLaMA-2 (Touvron et al., 2023)	LLaMA-2, decoder-only transformer	Excellent for NLG tasks, further fine-tuning needed for low-resource languages.
Phi-3 Mini (Abdin et al., 2024)	Phi-3 Mini for on-device deployment, causal LM	Efficient for offline inference, may sacrifice summary quality compared to larger models.

Table 15: Summary of related works. Models used in our work are highlighted in bold.

#Overlap	**Jaccard**	**Bigram**	**Trigram**	**BERTScore**
1	**0.1913**	**0.0657**	**0.0252**	**0.8682**
2	0.2055	0.0703	0.0261	0.8697
3	0.2269	0.0828	0.0345	0.8717

Table 16: Redundancy and semantic similarity metrics for different overlap strategies.

summary and the reference summary. It captures the basic vocabulary coverage and word-level similarity.

Formula:

$$R\text{-}1_{rec} = \frac{\sum_{gram_1 \in S_{ref}} C_{match}(gram_1)}{\sum_{gram_1 \in S_{ref}} C(gram_1)} \quad (4)$$

$$R\text{-}1_{pre} = \frac{\sum_{gram_1 \in S_{gen}} C_{match}(gram_1)}{\sum_{gram_1 \in S_{gen}} C(gram_1)} \quad (5)$$

$$R\text{-}1_{F1} = \frac{2 \times R\text{-}1_{pre} \times R\text{-}1_{rec}}{R\text{-}1_{pre} + R\text{-}1_{rec}} \quad (6)$$

where S_{ref} is the reference summary, S_{gen} is the generated summary, and $C_{match}(gram_1)$ represents the count of unigrams appearing in both summaries. $R-1$ is short for ROUGE-1. The subscripts pre, rec, and $F1$ mean precision, recall, and F1 score.

ROUGE-2: ROUGE-2 extends the evaluation to bigrams (two consecutive words). It captures phrase-level similarity and local coherence between summaries.

Formula:

$$R\text{-}2_{rec} = \frac{\sum_{gram_2 \in S_{ref}} C_{match}(gram_2)}{\sum_{gram_2 \in S_{ref}} Count(gram_2)} \quad (7)$$

$$R\text{-}2_{pre} = \frac{\sum_{gram_2 \in S_{gen}} Count_{match}(gram_2)}{\sum_{gram_2 \in S_{gen}} Count(gram_2)} \quad (8)$$

$$R\text{-}2_{F1} = \frac{2 \times R\text{-}2_{pre} \times R\text{-}2_{rec}}{R\text{-}2_{pre} + R\text{-}2_{rec}} \quad (9)$$

ROUGE-L: ROUGE-L measures the Longest Common Subsequence (LCS) between the generated and reference summaries. It captures sentence-level similarity without requiring consecutive matching.

Formula:

$$R\text{-}L_{rec} = \frac{LCS(S_{ref}, S_{gen})}{|S_{ref}|} \quad (10)$$

$$R\text{-}L_{pre} = \frac{LCS(S_{ref}, S_{gen})}{|S_{gen}|} \quad (11)$$

$$R\text{-}L_{F1} = \frac{2 \times R\text{-}L_{pre} \times R\text{-}L_{rec}}{R\text{-}L_{pre} + R\text{-}L_{rec}} \quad (12)$$

Algorithm 4 Pipeline of Bengali Academic Video Summarization with Title.

```
Require: Academic video file $V$
Ensure: Summary $S$ with descriptive title $T$
Audio Extraction:
$audio \leftarrow$ ExtractAudioStream$(V)$
Audio-to-Text Transcription:
$text_{raw} \leftarrow$ SpeechToText$(audio)$
Text Preprocessing:
$text_{punct} \leftarrow$ AddPunctuation$(text_{raw})$
Text Chunking:
$chunks \leftarrow$ SplitwOverlap$(text_{punct}, \text{overlap} = 1)$
Summary Generation:
for each $chunk_i$ in $chunks$ do
    $summary_i \leftarrow$ bT5.Generate$(chunk_i)$
end for
$S_{combined} \leftarrow$ Concatenate$(\{summary_i\})$
Timestamp Mapping:
$S_{timed} \leftarrow$ MapToTimestamps$(S_{combined}, V)$
Title Generation:
$T_{raw} \leftarrow$ mT5-multilingual-XLSum$(S_{combined})$
$T \leftarrow$ ApplyRegexProcessing$(T_{raw}, S_{timed})$
Output:
return Summary $S_{timed}$ with title $T$
```

where $LCS(S_{ref}, S_{gen})$ represents the length of the longest common subsequence between reference and generated summaries, and $|S|$ denotes the length of sequence S.

BERTScore: BERTScore uses pre-trained BERT embeddings to compute similarity scores between generated and reference summaries at the semantic level. It addresses limitations of lexical overlap metrics (Zhang et al., 2025). Unlike ROUGE metrics, BERTScore can capture semantic similarity even when different words are used to express similar meanings.

Formula: For tokens x_i in the reference summary and $\hat{x}_j$ in the generated summary:

$$BERTScore_{rec} = \frac{1}{|x|} \sum_{x_i \in x} \max_{\hat{x}_j \in \hat{x}} \mathbf{x}_i^T \hat{\mathbf{x}}_j \tag{13}$$

$$BERTScore_{pre} = \frac{1}{|\hat{x}|} \sum_{\hat{x}_j \in \hat{x}} \max_{x_i \in x} \mathbf{x}_i^T \hat{\mathbf{x}}_j \tag{14}$$

$$BS_{F1} = \frac{2 \times BS_{pre} \times BS_{rec}}{BS_{pre} + BS_{rec}} \tag{15}$$

where $\mathbf{x}_i$ and $\hat{\mathbf{x}}_j$ represent contextualized embeddings from BERT for tokens in reference and generated summaries, respectively. BS also represents BERTScore.

MoverScore: MoverScore addresses the limitation of BERTScore's greedy matching by employing Earth Mover's Distance to find the optimal alignment between tokens in generated and reference summaries (Zhang et al., 2025). This metric considers the global optimal transport between semantic representations.

Formula: MoverScore is computed as:

$$Mover = 1 - \frac{W_d(\mathbf{E}_{ref}, \mathbf{E}_{gen})}{\max(||\mathbf{E}_{ref}||_2, ||\mathbf{E}_{gen}||_2)} \tag{16}$$

where $W_d(\mathbf{E}_{ref}, \mathbf{E}_{gen})$ represents the Earth Mover's Distance between the embedding sets $\mathbf{E}_{ref}$ and $\mathbf{E}_{gen}$ of the reference and generated summaries, respectively, and $|| \cdot ||_2$ denotes the L2 norm.

Factual Consistency Metrics: Factual consistency evaluation is crucial for abstractive summarization systems to ensure that generated summaries do not introduce false information or contradict the source content (Zhang et al., 2025). We employ three approaches to assess factual accuracy.

FactCC: FactCC (Factual Consistency Checking) uses a BERT-based classifier trained specifically to detect factual inconsistencies between source documents and summaries (Zhang et al., 2025). The model outputs a binary classification score indicating whether the summary is factually consistent with the source.

Formula:

$$FactCC_{score} = P(con|doc, S) \tag{17}$$

where con, doc, and S are short for consistent, document, and summary, respectively. $P(con|doc, S)$ represents the probability that the summary is factually consistent with the source document as determined by the trained classifier.

SummaC: SummaC (Summary Consistency) evaluates factual consistency using multiple complementary approaches (Zhang et al., 2025). It combines natural language inference (NLI) models and question-answering systems to detect inconsistencies.

Formula:

$$SummaC = \frac{1}{N} \sum_{i=1}^{N} P(e|sent_i^S, doc) \tag{18}$$

where N is the number of sentences in the summary, e stands for entailment, $sent$ is short for sentence, S is short for summary, and doc represents document. and $P(e|sent_i^S, doc)$ represents the entailment probability for each summary sentence given the source document.

DAE: DAE (Dependency Arc Entailment) evaluates factual consistency by comparing dependency parsing structures between source and summary sentences. It focuses on grammatical relationships and semantic roles (Zhang et al., 2025).

Formula:

$$DAE_{score} = \frac{|Arcs_S \cap Arcs_{entailed}|}{|Arcs_S|} \qquad (19)$$

where S represents a summary. $Arcs_S$ represents the set of dependency arcs in the summary, and $Arcs_{entailed}$ represents the set of arcs that can be entailed from the source document.

Word Error Rate and Character Error Rate: WER and CER are standard metrics for evaluating ASR systems. WER measures the proportion of words in the reference transcription that are incorrectly predicted. It is calculated as:

$$\text{WER} = \frac{S + D + I}{N} \qquad (20)$$

where S is the number of substitutions, D is the number of deletions, I is the number of insertions, and N is the total number of words in the reference. Similarly, CER measures the proportion of characters incorrectly predicted, using the same formula but at the character level.

A.4 Parameters, Context Window, and Inference Time Comparison

Model	**P**	**C. W.**	**T (s)**
BanglaT5	247M	512	2.54
NLLB-200-Distilled	600M	512	3.76
mBART-50	610M	1024	2.78
mT5 (small)	300M	512	3.68
LLaVa v1.6 mistral	7000M	32K	44.59
Qwen 2.5-1.5B instruct	1500M	128K	78.32
Phi 3- mini 128K instruct	3800M	128K	12.34
Llama-2 7B instruct v01	7000M	2048	11.95

Table 17: Number of Parameters and Context Window Size Comparison.

The number of parameters, inference time and the context window size of the models used (for inference and fine-tuning) by the authors are provided in Table 17. *BanglaT5* is a lightweight model, but it has a noticeably smaller context window size compared to the recent models. Despite this, the model achieved a remarkable performance in the summary generation task, with the smallest inference time. It is to be noted that the inference time of the first four models is calculated after fine-tuning them. Since we have not fine-tuned the latter four, the inference time for them is calculated from the raw model. Time, T of the Table 17, is the time required by each model for generating 150 tokens, and is mentioned in seconds (s). P denotes the number of parameters, and C.W. indicates the context window size.

A.5 Summary Generation Results with Bootstrap

Table 5displays the bootstrapped means and half-widths for a 95% confidence interval. A more elaborate version of the data, with interval range and interval widths for ROUGE, BERT, and mover-scores, is presented in Table **??**.

Metric		**bT5**	**mT5-XL**
ROUGE-1 (F1)	Mean	0.1033	**0.4476**
	95% CI	[0.0658, 0.1466]	[0.3727, 0.5258]
	CI Width	0.0808	0.0.1531
ROUGE-2 (F1)	Mean	0.0136	**0.2129**
	95% CI	[0.0000, 0.0319]	[0.1416, 0.2918]
	CI Width	0.0319	0.1502
ROUGE-L (F1)	Mean	0.0942	**0.3720**
	95% CI	[0.0607, 0.1313]	[0.3016, 0.4483]
	CI Width	0.0705	0.1467

Table 18: Fine-tuned models comparison for title generation. CI denotes Confidence Interval, and bT5 stands for BanglaT5

A.6 Title Generation Results with Bootstrap

To complement the ROUGE precision, recall, and F1 scores reported in the main paper, we provide a more detailed statistical analysis of the title generation results. Table 18 reports the bootstrap means, 95% confidence intervals, and interval widths for the ROUGE F1 scores. These statistics offer a more robust understanding of model variability

Metric	Human Sum	Generated Sum	Reviewer
Grammatical Consistency	2.85	2.43	1
Information Retrieval	2.71	2.56	
Factual Consistency	2.76	2.50	
Conciseness	2.68	2.51	
Grammatical Consistency	2.79	2.27	2
Information Retrieval	2.89	2.37	
Factual Consistency	2.86	2.44	
Conciseness	2.59	2.47	
Grammatical Consistency	2.99	2.03	3
Information Retrieval	2.01	2.03	
Factual Consistency	3	2.91	
Conciseness	2	2.13	
Grammatical Consistency	2.94	2.16	4
Information Retrieval	2.86	2.32	
Factual Consistency	2.75	2.36	
Conciseness	2.67	2.23	
Grammatical Consistency	2.86	2.38	5
Information Retrieval	2.82	2.26	
Factual Consistency	2.76	2.42	
Conciseness	2.71	2.38	
Grammatical Consistency	2.82	2.11	6
Information Retrieval	2.85	2.12	
Factual Consistency	2.72	2.07	
Conciseness	2.65	2.27	
Grammatical Consistency	2.86	2.21	7
Information Retrieval	2.74	2.46	
Factual Consistency	2.62	2.50	
Conciseness	2.77	2.82	

Table 19: Average Likert scores by reviewer for summary evaluation (for 100 samples).

and reliability across different samples. It ensures the consistency of the improvements observed.

A.7 Likert Score

Table 7 presents the average overall Likert score of the 100 samples, while Table 19 reports the average Likert scores for all the samples evaluated by each reviewer. Here, Sum is short for summary.

A.8 Impact of ASR Choice on End-to-End Video Summarization

To examine the effect of the transcription module on downstream performance, we conduct an end-to-end evaluation of the Bengali video summarization pipeline by comparing the Google's speech recognition API used in our final system with the second-best open-source transcription model, facebook/mms-1b-all (identified in Table 4), and report the resulting performance in Table 20. The results show that although the open-source Facebook MMS model produces competitive summaries, the pipeline using Google ASR consistently outperforms it across ROUGE, BERTScore, and MoverScore metrics. This highlights the importance of transcription quality for abstractive summarization, particularly for low-resource languages such as Bengali.

Metric	Facebook MMS	Google ASR (Proposed)
ROUGE-1 (F1)	0.4609	**0.5275**
ROUGE-2 (F1)	0.1590	**0.2025**
ROUGE-L (F1)	0.1753	**0.2211**
BERTScore (F1)	0.8736	**0.8838**
MoverScore	0.5190	**0.5649**

Table 20: Comparison of end-to-end summarization performance using different transcription models.

To demonstrate the flexibility of our framework, we replace the Google ASR API with a strong open-source transcription model, facebook/mms-1b-all (Facebook MMS), while keeping all subsequent stages in the video summarization pipeline unchanged. The proposed video summarization pipeline is modular and adaptable, allowing the ASR component to be replaced with any suitable open-source or fine-tuned transcription model. We select Google ASR for the final system due to its superior performance on Bengali speech. However, future researchers may substitute it with alternative ASR models as stronger open-source systems become available or when a fully open pipeline is required.

A.9 Qualitative Results

For language generation tasks, qualitative results are of utmost importance. The Appendix sections A.9.1, A.9.2, A.9.3, and A.9.4 show the qualitative results for audio transcription, summary generation, title generation, and final output, respectively.

A.9.1 Audio Transcription

The qualitative result for Audio Transcription, mentioned in section 4.1, is tabulated in Table 21. This shows the transcription improvement after pre-processing the audio.

A.9.2 Summary

Text is the transcription obtained from the video after the audio extraction, pre-processing, and punctuation adding steps. *Expected Summary* is a human-generated summary that is used as a reference summary. The *Generated Summaries* are the summaries generated by the four fine-tuned models for the text mentioned.

Before Enhancement	After Enhancement
আমরা অনেক সময় যেটা করি সেটা হচ্ছে যে লজিক সার্কিট এটাকে অবজার্ভ করি অর্থাৎ আমরা হচ্ছে বিভিন্ন ইনপুটের জন্য দেখি যে আসলে এটার আউটপুট কি তো এটাকে হচ্ছে আমরা একটা **চার টাকার এক্সপ্রেস** করি যেটাকে হচ্ছে আমরা বলি ট্রুথ টেবিল বা সত্যক সারণি আমাদের অনেক সময় সত্যক সারণি বাদে আর কিছু দেয়া থাকে না ওই লজিক সার্কিট এর ব্যাপারে শুধুমাত্র আমাদের হচ্ছে সত্যক সারণি থেকে ধারণা নিতে হয়	আমরা অনেক সময় যেটা করি সেটা হচ্ছে লজিক সার্কিট এটাকে অবজার্ভ করি অর্থাৎ আমরা হচ্ছে বিভিন্ন ইনপুটের জন্য দেখি যে আসলে এটার আউটপুট কি তো এটাকে হচ্ছে আমরা একটা **চার্ট আকারে এক্সপ্রেস** করি যেটাকে হচ্ছে আমরা বলি ট্রুথ টেবিল বা সত্যক সারণি **তো** আমাদের অনেক সময় সত্যক সারণি বাদে আর কিছু দেয়া থাকে না ওই লজিক সার্কিট এর ব্যাপারে শুধুমাত্র আমাদের হচ্ছে সত্যক সারণি থেকে ধারণা নিতে হয়

Table 21: Transcription Comparison (Before and After Enhancement). Bold words are transcribed accurately after preprocessing, which were incorrectly transcribed Before Enhancement.

Text: তাইলে যদি এই টোটাল দৈর্ঘ্যটা আমাদের এল হয় তাইলে এই দৈর্ঘ্যটাও এল হবে। তার কারণ এই দোলকটাই দুলতে দুলতে এখানে চলে গেছে। দৈর্ঘ্য তো সেম। যদি এটা এক্স হয় তাহলে এটার মান কিন্তু আমরা বের করতে পারি। এতটুকুর মান আমরা বের করতে পারি। এতটুকুর মান আমরা বের করতে পারি। এতটুকুর মান কি হবে ভাই? এতটুকুর মান হবে। আশা করি বুঝতেছো যে আমার যদি এখানে এইচ হয়, এইচ কিন্তু দেওয়া আছে। এইচ টাকে নিতে হবে। এতটুকু হচ্ছে আমার এইচ। আর টোটালে যদি এল হয় তাহলে এইচ টুকু বাদ দিলে এটা হবে এল মাইনাস এইচ। এটা কি হবে? এল মাইনাস এইচ। আচ্ছা ফাইন। এখন যদি আমি শুধুমাত্র যে এতটুকুতে পিথাগোরাস এপ্লাই করি আমার মান লাগবে কার? এক্স এর লাগবে। তো এজন্য পিথাগোরাস এপ্লাই করতেছি। প্রশ্নে কিন্তু আমাকে দেখো কি কি দেওয়া আছে? এল দেওয়া আছে। তারপর হচ্ছে তোমার এইচ দেওয়া আছে। মানে প্রশ্নের এই দুইটা মান আমি জানি। তাহলে চিত্র থেকে এই এল এর মান আমি জানি আবার এইচ এর মানও জানি। তার মানে এল মাইনাস এইচ ও জানি। শুধুমাত্র জানিনা কোনটা? এক্স জানিনা। তাহলে এক্স এর মানটা আমাকে বের করতে হবে। পিথাগোরাস যদি এপ্লাই করি তাহলে অতিভুজ স্কয়ার হবে ভূমি স্কয়ার প্লাস লম্ব স্কয়ার। তাহলে এখানে লিখলা র একটা নাম দেই। হ্যাঁ, সহজে বোঝার স্বার্থে আর কি। ত্রিভুজের নাম দিই অন্য কালি দিয়ে। এটা হচ্ছে। ধরো এ বি সি এটাকে। তাহলে আমরা একটা কাজ করি। এটাকে সাইডে রাখি। তাহলে এবি সি ত্রিভুজে দুইবার লিখলাম। ত্রিভুজ বাহুল্য হয়ে গেল। কোন প্যারা নাই। বাংলা ক্লাস তো আর না। এল স্কয়ার সমান এল মাইনাস এইচ হোল স্কয়ার প্লাস হচ্ছে এক্স স্কয়ার। এবার তাইলে সূত্রটা ভেঙে ফেলি। এল স্কয়ার মাইনাস টুয়াইস এল এইচ প্লাস এইচ স্কয়ার প্লাস হচ্ছে এক্স স্কয়ার। দেখো তো এখানে এই যে এল স্কয়ার এল স্কয়ার যদি বাদ হয় তাহলে আমি এখান থেকে বলতে পারবো এক্স স্কয়ার এর মান হবে টুয়াইস এল এইচ মাইনাস এইচ স্কয়ার। দ্যাট মিন্স এক্স এর ভ্যালু হবে আন্ডার রুট টুয়াইস এল এইচ মাইনাস এক্স স্কয়ার এইচ স্কয়ার। এই এক্স এর মানটা আমি এখানে বসায় দিব। ঠিক আছে? কারণ এক্স এর মান আমরা জানিনা। চিনি না। এক্স কে চিনি না। এজন্য এক্স এর মান আমি এখানে রাখবো না।

Expected Summary: দোলকের উপর বাতাসের ক্রিয়া এবং তার উচ্চতার পরিবর্তন নিয়ে আলোচনায়, প্রথমে দৈর্ঘ্য এল এবং উচ্চতা এইচ দেওয়া থাকে। এখান থেকে পিথাগোরাসের থিওরি প্রয়োগ করে, এইচ বাদ দিলে অবশিষ্ট দৈর্ঘ্য এল মাইনাস এইচ পাওয়া যায়। পিথাগোরাস প্রয়োগের মাধ্যমে অতিভুজ স্কোয়ার সমান ভূমি স্কোয়ার এবং লম্ব স্কোয়ারের যোগফল হিসেবে সূত্রটি ভেঙে দেওয়া হয়। পরে, এক্স এর মান বের করতে উল্লিখিত সূত্রে এক্স স্কোয়ারের মান পাওয়া যায়। এক্স এর মানের জন্য শেষ সূত্র হিসেবে আন্ডাররুট টুয়াইস এল এইচ মাইনাস এইচ স্কোয়ার প্রাপ্ত হয়।

Generated Summary (*BanglaT5*): এই অংশে পিথাগোরাসের সূত্র ব্যবহার করে অতিভুজ স্কয়ার এবং লম্ব স্কয়ারের মান বের করার প্রক্রিয়া ব্যাখ্যা করা হয়েছে। প্রথমে, অতিভুজ স্কয়ার ও লম্ব স্কয়ার এর মান বের করার জন্য পিথাগোরাস প্রয়োগ করা হয়েছে, যেখানে অতিভুজ স্কয়ার ভূমি স্কয়ার প্লাস লম্ব স্কয়ার। এরপর, ত্রিভুজের নাম পরিবর্তন করে এক্স এর মান বের করা হয়েছে।

Generated Summary (NLLB): এই অংশে বক্তা ত্রিভুজ সম্পর্কিত একটি গাণিতিক বিশ্লেষণ করছেন, যেখানে দুটি বাহুর দৈর্ঘ্য (এল, এল, এবং এইচ) দেওয়া আছে এবং তাদের যোগফল (এইচ) বের করার জন্য এল মাইনাস এইচ সূত্র ব্যবহার করা হচ্ছে। এরপর, তিনি ত্রিভুজের দুটি বাহুর যোগফল (এল স্কয়ার মাইনাস এল স্কয়ার প্লাস এইচ স্কয়ার) বের করার প্রক্রিয়া ব্যাখ্যা করছেন, যেখানে এল স্কয়ার এবং এইচ মাইনাস লম্বের যোগফল দেওয়া আছে। বক্তা বলছেন যে, এইচ প্লাস এল স্কয়ার সমীকরণটি সঠিকভাবে প্রয়োগ করতে হবে, এবং এই সূত্রটি সঠিকভাবে ব্যবহার করে সঠিক ফলাফল পাওয়া যাবে।

Generated Summary (mBART): এখানে দোলকের দৈর্ঘ্য বের করার জন্য পিথাগোরাস প্রয়োগ করা হয়েছে, যেখানে এল এবং এইচ এর মান জানা ছিল। চিত্র থেকে দেখা যাচ্ছে যে, যদি এক্স হয়, তাহলে এক্স-এর মান বের করতে হবে। এরপর, অতিভুজ স্কয়ার সূত্র ব্যবহার করে ত্রিভুজের স্থানাঙ্ক নির্ণয় করা হয়। সূত্রটি ভেঙে ফেলা হয় এবং একটি নতুন সমীকরণ পাওয়া যায়, যা হলো: (এল2 - ২এলএইচ + এক্স2 +

২এবিসি)। এই সূত্রের মাধ্যমে, এক্স এবং অন্যান্য মান নির্ধারণ করা সম্ভব।

Generated Summary (mT5): এই অংশে পিথাগোরাস এপ্লাই করার প্রক্রিয়া ব্যাখ্যা করা হয়েছে। প্রথমে, যদি টোটাল দৈর্ঘ্য এল হয়, তাহলে এইচ এর মান হবে এল মাইনাস এইচ। এরপর, ত্রিভুজ স্কয়ার প্লাস লম্ব স্কয়ার এবং অতিভুজের নাম দেওয়া হয়েছে, যেখানে ত্রিভুজের নাম নির্ধারণ করা হয়েছে এবং এরপর ত্রিভুজে ভূমি স্কয়ার হবে ভূমি, এবং এটি ভূমি। এই প্রক্রিয়ায়, যদি এল হয়, তবে এই সূত্র ব্যবহার করা হয়েছে, যেমন এ বি সি, এ বি সি এবং বি সি এর মধ্যে একটি সূত্র দেওয়া হয়েছে।

The fine-tuned *BanglaT5* model generates a more accurate and precise summary, which is useful for the academic domain. Here, all the summaries are inferred from the fine-tuned models with our hand-made dataset.

A.9.3 Title

Expected Title: পরমাণু মডেল এবং আলো: তরঙ্গ এবং কণার বৈশিষ্ট্য

Generated Title (raw *mT5-multilingual-XLSum*): যুক্তরাষ্ট্রের পরমাণু গবেষণা ইন্সটিটিউটের দ্বিতীয় লেকচারে নতুন পর

Generated Title (fine-tuned *mT5-multilingual-XLSum*): পরমাণু মডেল: কোয়ান্টাম থিওরি এবং তরঙ্গ সমীকরণ

Generated Title (fine-tuned *BanglaT5*): প্লাঙ্কের কোয়ান্টাম তত্ত্বে আলোর তরঙ্গ তত্ত্ব নিয়ে কাজ করেছিলেন। প্লাঙ্কের কোয়ান্টাম তত্ত্বে আলোর তরঙ্গ তত্ত্ব নিয়ে কাজ

A.9.4 Example of Final Output

The final output format includes the generated title, compression ratio, generated summaries, and added timestamps.

Title: টাইম শেয়ারিং এবং মাল্টিটাস্কিং অপারেটিং সিস্টেম: জব এক্সিকিউট প্রক্রিয়া

Compression Ratio: 0.2276

[00:00 - 02:53] এখানে টাইম শেয়ারিং অপারেটিং সিস্টেম এবং মাল্টিটাস্কিং অপারেটিং সিস্টেমের মধ্যে পার্থক্য আলোচনা করা হয়েছে। ব্যাচ অপারেটিং সিস্টেমে একটি মাত্র জব এক্সিকিউট করা হয়, যার জন্য সিপিইউ আইডল থাকে না, এবং এটি পরবর্তী প্রসেসের জন্য ব্লক করে দেওয়া হয়। এই প্রক্রিয়ায়, সিপিউ আইডেল থাকে না এবং এটি একটি পোস্ট কেস হিসেবে কাজ করে। এরপর, যখন জব এক্সকিউট হয়, তখন সিপিইউ ব্লক করে পরবর্তী প্রসেসে চলে যায়। এই প্রক্রিয়ার মাধ্যমে, সিপিইউ আইডিলনেস কাটানো হয় এবং পরবর্তী প্রসেসগুলোর জন্য আরও বেশি সময় লাগে।

[02:47 - 05:15] শিক্ষক একটি জব ট্রেন সম্পর্কে আলোচনা করছেন, যেখানে পাঁচটি জবের উত্তর দেওয়া হয়েছে, যার মধ্যে দুটি প্রশ্নের উত্তর দিতে প্রায় থেকে মিনিট সময় লেগেছে। এরপর, শিক্ষক স্টুডেন্টদের কাছে প্রশ্ন করেন, তাদের উত্তর দেওয়ার জন্য প্রায় থেকে 3 মিনিট সময় লাগে, কিন্তু কোনো উত্তর দেওয়া হয়নি। এরপর শিক্ষক বলেন, জব ট্রেন আগে উত্তর দিতে পারে, কিন্তু আউটপুট অপারেশন করার জন্য তার কোন ইনপুট আউটপুট যাওয়ার প্রয়োজন নেই, তাই তিনি বলেন যে, জবটি দ্রুত সম্পন্ন হতে পারে, কারণ তার কোনো ইনপুট আউটপুটের প্রয়োজন নেই।