Multilingual Speech-to-Text and Speech-to-Speech Content Summarization for Indian Languages

Arnav Sharma Himanshu
Department of Computer Science
Indian Institute of Technology, Jodhpur
m24cse004@iitj.ac.in, m24cse009@iitj.ac.in

Abstract

This report presents a multilingual speech-to-speech summarization pipeline that processes raw audio inputs into concise, natural-sounding spoken summaries. The system integrates multiple stages, including audio preprocessing, automatic speech recognition (ASR), text summarization, language detection, speaker selection, and text-to-speech (TTS) synthesis. Leveraging the Whisper model for ASR and fine-tuning it on task-specific data significantly reduced both Word Error Rate (WER) and Character Error Rate (CER), enhancing transcription accuracy. The mT5 model was used for summarization, where fine-tuning resulted in improved ROUGE scores compared to the pretrained baseline, reflecting better content compression and relevance. To address the lack of large-scale supervised datasets, a synthetic data generation module was developed using the Indic Parler-TTS model , enabling the creation of high-quality audio-text pairs in multiple Indian languages. The pipeline ensures consistency between language and speaker identity through automatic language detection and voice mapping. The end-to-end system demonstrates strong performance and fluency across all components, highlighting its potential for applications in multilingual media summarization, voice assistants, and educational tools.

1. Introduction

In today's digital world, spoken content is being produced and consumed at an unprecedented rate. From corporate meetings and classroom lectures to podcasts, news broadcasts, and interviews, audio-based information has become a vital medium for communication and knowledge sharing. However, navigating through lengthy spoken material can often be inefficient and time-consuming, especially when quick comprehension or information retrieval is needed. This is where speech summarization proves to be immensely valuable [7]. By condensing spoken content into shorter, coherent versions without losing its essential meaning, speech summarization enables faster consumption, enhances accessibility, improves indexing, and facilitates real-time understanding of long-form audio. It holds significant importance across a range of sectors such as education, journalism, healthcare, legal documentation, and customer service.

Despite the evident utility and potential of automatic speech summarization, the majority of existing systems are designed for monolingual, high-resource language settings — primarily English. Most state-of-the-art pipelines are developed and fine-tuned using large, richly annotated datasets that are typically available only for a handful of globally dominant languages. This has led to a technological disparity for linguistically rich but underrepresented regions like India, where there is a scarcity of comprehensive solutions that can handle the diverse linguistic ecosystem. Although considerable progress has been made in the fields of automatic speech recognition (ASR) and text-to-speech synthesis (TTS) for Indian languages, these developments often remain isolated. There is a lack of integrated, end-to-end systems that can take in raw spoken input and output a concise spoken summary, particularly across multiple Indian languages.

India's linguistic landscape, comprising 22 officially recognized languages and hundreds of regional dialects, poses unique challenges but also underscores the necessity for inclusive, scalable solutions [5]. In multilingual settings such as government institutions, educational setups, media organizations, and grassroots-level interactions, individuals predominantly communicate in their native languages. These languages, however, are frequently underserved by the existing technological ecosystem. This motivates the development of tools that not only recognize and transcribe speech in various Indian languages but also summarize and vocalize the content effectively, making information more accessible to a broader population.

A significant bottleneck in advancing this area of research is the limited availability of high-quality datasets

for speech-based summarization, especially for Indian languages. While many benchmark datasets exist for text-to-text summarization tasks, there is a noticeable absence of publicly available resources that pair speech inputs with corresponding summaries. Even more scarce are datasets that offer this alignment in low-resource Indian languages. To bridge this gap, our work contributes by generating a synthetic dataset derived from the XLSum dataset [4], a large multilingual corpus for text summarization. We augment this dataset by converting the textual data into speech using a TTS model, thereby creating paired speech-to-text and speech-to-speech summarization examples for eight Indian languages. This synthetic corpus not only aids in the training and evaluation of our proposed pipeline but also provides a valuable resource for the broader research community working in this domain.

To address the broader challenge of multilingual summarization, this project introduces a speech-to-speech summarization pipeline tailored specifically for Indian languages. The proposed system takes raw audio input, processes it through three main stages, and outputs a spoken summary. It begins with transcription using a fine-tuned Whisper ASR model [6], followed by summarization through a multilingual mT5 model [10] that extracts the most relevant information in a concise form. Finally, the summary is converted back into speech using Indic Parler-TTS [1], producing natural-sounding audio in the same language as the input. The system is capable of detecting the input language and automatically selecting the appropriate voice for synthesis, ensuring a smooth and contextually accurate output.

This work leverages state-of-the-art, open-source components and integrates them into a cohesive, modular, and scalable architecture. It stands as a prototype that demonstrates the feasibility and effectiveness of end-to-end speech summarization across multiple Indian languages. For the scope of this study, the system currently supports eight languages — English, Hindi, Punjabi, Urdu, Bengali, Tamil, Telugu, and Marathi. However, the modular design and generalizability of the approach ensure that additional languages can be incorporated in the future by following the same methodology. By doing so, this project contributes meaningfully to the broader effort of making speech technologies more inclusive, equitable, and accessible across India's diverse linguistic landscape.

2. Literature Review

In 2023, Urlana et al. [9]introduced PMIndiaSum, a large multilingual and cross-lingual summarization dataset covering 14 Indian languages and 196 language pairs, built from the Prime Minister of India's website. It included 76K monolingual and 620K cross-lingual document-headline pairs. The authors ensured data quality, copyright compliance, and broad language coverage. They evaluated it using fine-tuning, prompting, and translation-based methods, showing its usefulness for Indian language summarization. The dataset was released publicly for research use. the work Taunk and Varma [8] explored abstractive summarization of Indian languages using multilingual transformer models, specifically IndicBART and mT5, as part of the FIRE ILSUM 2022 shared task. The authors used the provided dataset, applied data augmentation, and evaluated performance using ROUGE-1 to ROUGE-4 scores. Their work highlighted the growing relevance of multilingual models for low-resource Indian languages and the need for more summarization datasets.in 2021 Bai et al . [2] proposed MCLAS, a novel multi-task framework for Cross-Lingual Abstractive Summarization (CLS) in low-resource settings. Unlike prior methods using separate decoders, MCLAS employed a shared decoder to jointly learn monolingual and cross-lingual summarization by generating concatenated outputs. This approach enabled better knowledge transfer between high- and low-resource languages. Experiments on two CLS datasets showed that MCLAS outperformed baseline models, and analysis confirmed that it effectively captured cross-lingual interactions despite limited parallel data. Hasan et al. introduced XL-Sum, a large-scale multilingual abstractive summarization dataset containing 1 million article-summary pairs in 44 languages, sourced from the BBC. It included many low-resource languages with no prior datasets. The authors fine-tuned mT5 on XL-Sum and achieved strong results in both multilingual and low-resource settings, with ROUGE-2 scores exceeding 11–15 on several languages. XL-Sum was noted for its quality and scale, and the dataset and models were publicly released to support future research.

Chadha et al. introduced Vakyansh, an end-to-end ASR toolkit for low-resource Indic languages. It automated data creation, model training, evaluation, and deployment. The authors created 14,000 hours of speech data across 23 languages, trained wav2vec 2.0-based models, and fine-tuned them for 18 Indic languages. They also included language models and punctuation restoration. All resources were open-sourced to support speech-based application development in Indic languages. Gupta et al. introduced CLSRIL-23, a self-supervised multilingual audio model trained on 10,000 hours of raw speech across 23 Indic languages using wav2vec 2.0. It learned shared cross-lingual representations by solving a contrastive task over masked speech latents.

3. Dataset

To build an effective and multilingual speech-to-speech summarization pipeline, two distinct datasets were utilized — one for fine-tuning the automatic speech recognition (ASR) model and the other for training and evaluating the text summarization module. The selection of datasets was guided by the need for high-quality, diverse, and language-

rich resources that cover multiple Indian languages.

The Google FLEURS dataset [3] was employed for fine-tuning the Whisper ASR model. FLEURS (Few-shot Learning Evaluation of Universal Representations of Speech) is a multilingual dataset developed by Google, aimed at benchmarking ASR and speech representation learning models across a wide range of languages. It contains read speech data aligned with transcriptions, carefully curated to support tasks like language identification, speech recognition, and speech understanding. For this project, a subset of the FLEURS dataset corresponding to the eight target languages — English, Hindi, Punjabi, Urdu, Bengali, Tamil, Telugu, and Marathi — was extracted and used for fine-tuning Whisper. The dataset offers consistent quality across languages and provides a balanced distribution of audio samples, making it suitable for developing robust multilingual ASR capabilities.

For the summarization component, the XL-Sum dataset [4] was used. Developed by the Centre for Speech and Language Technologies at CSEBUETNLP, XL-Sum (Cross-Lingual Abstractive Summarization) is a large-scale multilingual dataset specifically designed for text summarization. It consists of BBC news articles and corresponding single-sentence summaries in 45 different languages, including all eight languages targeted in this project. The dataset is well-suited for training models like mT5 for abstractive summarization due to its clean, real-world news content and the availability of parallel summaries across languages. Given the lack of publicly available Indian language datasets for speech-to-text summarization, XL-Sum served as the foundation for generating a synthetic dataset to simulate this task.

By combining Google FLEURS for ASR fine-tuning and XL-Sum for summarization, the project effectively leverages high-quality multilingual resources to train and evaluate each component of the pipeline. This data-driven approach ensures that the system is capable of performing accurate transcription and meaningful summarization across diverse Indian languages.

4. Methodology

In this project, we propose an end-to-end multilingual speech-to-speech summarization pipeline tailored for Indian languages. The methodology consists of three core components: (1) transcription of spoken audio into text using an Automatic Speech Recognition (ASR) model, (2) summarization of transcribed text using a multilingual text-to-text transformer, and (3) conversion of the generated summary into natural-sounding speech through a text-to-speech (TTS) model. Each component has been carefully trained or fine-tuned using open-source datasets and models, allowing the entire system to handle diverse Indian languages efficiently. This section explains the implementation and training details of the ASR and summarization modules used in the pipeline.

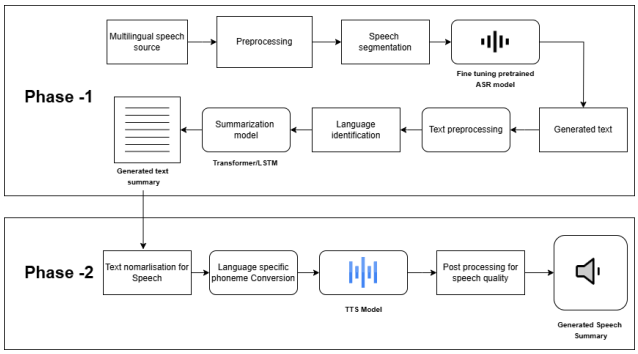

Figure 1. Proposed Methodology for Multilingual Speech-to-Text and Speech-to-Speech Content Summarization

4.1. Automatic Speech Recognition (ASR) Model

The first stage in the pipeline is automatic speech recognition (ASR), which involves converting spoken input into textual transcripts. For this purpose, we employed OpenAI's Whisper-small model, a robust multilingual ASR system known for its generalization capabilities. In order to adapt the model more effectively to Indian language inputs, we fine-tuned it on the Google FLEURS dataset, which provides transcribed speech data for a wide array of languages, including the eight selected for this study: English, Hindi, Punjabi, Urdu, Bengali, Tamil, Telugu, and Marathi. Each of these language-specific subsets was loaded and combined to form a comprehensive multilingual training dataset. This combined dataset was then split into training and test subsets, ensuring that the model was evaluated fairly across all languages.

Preprocessing involved converting the raw audio waveforms into feature representations using the WhisperProcessor, which includes both a feature extractor for audio and a tokenizer for textual transcriptions. The transcriptions were tokenized and padded appropriately to ensure consistency across batches. A custom data collator was implemented to manage this preprocessing step efficiently and prepare batches for model training.

Fine-tuning was conducted using the Hugging Face Trainer API, with the model trained for three epochs using a batch size of two and mixed precision enabled where supported. Loss values were tracked throughout training to monitor convergence. After fine-tuning, the model was evaluated using the Word Error Rate (WER) and Character Error Rate (CER) metrics, computed via the jiwer library. These metrics provided a quantitative measure of transcription quality by comparing model outputs with ground truth transcriptions in the test set.

To validate the effectiveness of fine-tuning, the perfor-

mance of the adapted Whisper model was compared with that of the original, unmodified Whisper-small model on a representative subset of the test data. The fine-tuned model consistently outperformed the baseline in both WER and CER across all evaluated languages, demonstrating enhanced accuracy in capturing linguistic nuances, dialectal variations, and pronunciation patterns characteristic of Indian languages. These improvements affirm the utility of language-specific fine-tuning for ASR tasks in multilingual settings, particularly when applied to underrepresented languages.

The outputs of this ASR stage, in the form of high-quality textual transcripts, serve as the input to the next component of the system: text summarization. This stage is described in detail in the following section. Please share the summarization code so we can continue building out the methodology accordingly.

4.2. Summarization Model

The second component of our pipeline casts transcription outputs into concise, informative summaries. To this end, we employ the XLSum corpus, a multilingual text-to-text summarization dataset covering our eight target languages. For each language, we extract a subset of examples, prepend a uniform "summarize:" prompt to the source text, and pair it with the provided human-written summary as the learning target. These per-language subsets are then merged, shuffled, and partitioned into training and validation splits to ensure that the model learns both language-specific and cross-lingual summarization patterns.

We build upon Google's multilingual T5 ("mT5-small") encoder–decoder architecture, initializing it from pretrained weights. Inputs are tokenized to a fixed length (up to 512 tokens) and padded or truncated as necessary, while targets are limited to 64 tokens to encourage brevity. During fine-tuning, we optimize the model using teacher-forced cross-entropy loss and evaluate generation quality by producing sequences via beam search. At the end of each epoch, we compute ROUGE-1, ROUGE-2, and ROUGE-L scores on the held-out validation set to monitor progress.

Training is carried out for multiple epochs with mixed-precision on GPU, using a moderate learning rate and gradient accumulation to balance learning stability and computation. After convergence, we visualize the loss trajectory over time to verify stable learning dynamics. Finally, we evaluate the fully fine-tuned model on the test split, reporting both numeric ROUGE metrics and qualitative examples of generated summaries. The result is an end-to-end text summarization module that, when chained with our ASR and TTS components, enables fully automated multilingual speech-to-speech summarization.

4.3. End-to-End Speech-to-Speech Summarization Pipeline

This section outlines the complete pipeline used to process speech input and generate an audio summary. The process involves multiple stages, starting with the preprocessing of audio data, followed by automatic speech recognition (ASR), text summarization, language detection, speaker selection, and finally, text-to-speech (TTS) synthesis. Each step is designed to ensure that the system delivers accurate, coherent, and contextually appropriate outputs. Below is a detailed explanation of each stage of the pipeline.

4.3.1 Audio Preprocessing

Before starting the transcription process, the raw audio undergoes several preprocessing steps to ensure it is in an optimal format for the Automatic Speech Recognition (ASR) model. The first task is normalizing the audio by ensuring it has only a single channel. Raw audio files might be recorded in stereo (two channels), but the ASR model works more efficiently with mono (one channel). To convert multi-channel audio into mono, the audio is averaged across both channels. Afterward, the sample rate of the audio is adjusted to 16 kHz, as the Whisper ASR model was trained with audio sampled at this rate. This step is crucial because different sample rates can affect the model's ability to correctly process the audio.

Another important step in audio preprocessing is noise reduction. Real-world audio often contains background noise, such as static, hum, or hiss, which can interfere with the transcription accuracy. To combat this, a spectral noise-reduction algorithm is applied. This algorithm identifies and attenuates background noise while preserving the main speech signal. Finally, to ensure that the transcription is free of irrelevant sounds, a Voice Activity Detection (VAD) filter is used. This filter removes leading and trailing silence in the audio, focusing the model on the sections of the audio that actually contain speech. As a result, the audio that enters the transcription model is clean, uniformly sampled, and optimized for further processing.

4.3.2 Automatic Speech Recognition (ASR)

The preprocessed audio is then passed through the ASR model, where the goal is to convert spoken words into text. However, long audio recordings can pose challenges due to input length limitations in most ASR models. To handle this, the audio is divided into smaller, overlapping chunks—typically 30 seconds long with a 3-second overlap. This chunking process helps the model handle long recordings by breaking them into manageable sections while still preserving context between chunks.

Each chunk is then passed through the Whisper feature extractor, which transforms the raw audio waveform into mel-spectrogram features. A mel-spectrogram represents the frequency content of the audio over time and is a crucial input for the ASR model. The ASR model processes these features and generates token logits, which are essentially probabilities for each possible token (word or subword) in the audio. The logits are then decoded into text using greedy generation, a method where the most likely token at each step is selected.

Once each chunk has been transcribed, the individual transcriptions are concatenated into a single, continuous transcription of the entire audio. This step ensures that long recordings, which might span multiple chunks, are accurately transcribed without losing context between them. The result is a seamless text representation of the original audio.

4.3.3 Text Summarization

After the transcription is complete, the next step is to generate a concise summary of the text. To do this, the full transcription is passed to the mT5 model, a multilingual encoder-decoder model trained to generate summaries. First, the transcription is tokenized, meaning it is split into smaller units (tokens), and then the text is padded and truncated to fit within the model's input length limits. This ensures that the input is consistent and does not exceed the model's processing capacity.

To determine the length of the summary, a dynamic heuristic is applied. This heuristic considers the length of the original transcription and a user-defined compression ratio. Based on this, the model adjusts how many new tokens it will generate, effectively controlling the length of the summary. The use of beam search during generation helps find the most coherent and informative sequence of words, while a length penalty ensures the summary is neither too long nor too short. For longer transcriptions, the summary might be more detailed, while for shorter transcriptions, it could be more concise.

The final result is a readable, human-friendly summary that retains the essential points of the original transcription, making it easier for users to quickly understand the content.

4.3.4 Language Detection and Speaker Selection

Before generating the speech output, the system needs to determine the language of the transcription and select an appropriate speaker to ensure the synthesized voice matches the language of the original speech. This step begins with language detection, where a language identification tool analyzes the transcription to infer the language. The system then maps the detected language code (such as "en" for English or "hi" for Hindi) to a predefined set of speaker profiles.

For example, if the language is detected as Hindi, the system would select the speaker "Rohit," whose voice is pre-configured for Hindi. This mapping is essential because different languages often require different phonetic characteristics and intonations, and selecting the correct speaker ensures the synthesized speech sounds natural. Additionally, by automatically choosing the correct speaker, the system removes the need for user intervention, simplifying the process while ensuring the output is accurate and consistent.

4.3.5 Text-to-Speech Synthesis

The final step in the pipeline involves converting both the full transcription and the summary into audio using text-to-speech (TTS) synthesis. The transcription and summary are first divided into smaller chunks, as TTS models typically have input size limitations. This ensures that the text can be processed without exceeding the model's maximum length.

Each chunk is then passed to the Indic Parler TTS model, which generates the corresponding audio waveform. During this step, a descriptive prompt is provided to the TTS model, specifying the speaker and language to use for the synthesis. The model then generates the corresponding audio for each chunk of text, and the individual audio segments are concatenated into two final output audio files: one for the full transcription and one for the summary.

The generated audio files are saved in WAV format, which can be played back by the user. These audio files allow the user to listen to both the original transcription and its concise summary, creating an engaging and efficient way to consume the content. Additionally, the system provides an option to play back the synthesized audio, allowing users to instantly hear the results of the processing pipeline.

4.4. Synthetic Data Generation

In many speech-to-speech applications, access to large, high-quality parallel corpora—where audio recordings are paired with corresponding text or summaries—is often limited, particularly for low-resource languages. To address this gap, we created a synthetic multilingual speech dataset by leveraging an existing text summarization resource and a state-of-the-art text-to-speech model. Starting from the XLSum corpus's article–summary pairs in eight target languages, we first mapped each language to a representative speaker identity (for example, "Rohit" for Hindi or "Arjun" for Bengali) and generated a brief descriptive prompt to characterize the desired voice profile ("Rohit's voice in Hindi is clear, moderately paced, and expressive").

Each article was split into suitably sized chunks to ensure compatibility with the TTS model's input length constraints. For each chunk, the descriptive prompt and chunk

text were tokenized and passed to the pretrained Indic Parler-TTS model, which produced waveform segments at the model's native sampling rate. These segments were then concatenated into a single continuous audio file representing the entire article. In parallel, the summary text underwent the same synthesis procedure—prompt formulation, tokenization, waveform generation, and concatenation—to yield a concise summary recording. Both text files (article and summary) and their corresponding audio files were saved together in a structured directory for each sample.

By converting high-quality text summaries into natural speech across multiple languages, this synthetic dataset serves as a versatile resource for training and evaluating end-to-end speech summarization systems. The resulting corpus not only fills a critical data gap for Indian languages but also provides tightly aligned text-and-speech pairs for both long-form content and its condensed summaries, thereby enabling future research in multilingual speech processing and cross-modal generation.

5. Results

5.1. Evaluation Metrics

To assess the performance of the proposed speech summarization pipeline, we use three primary evaluation metrics: ROUGE for summarization quality, and Word Error Rate (WER) and Character Error Rate (CER) for transcription accuracy.

5.1.1 Word Error Rate

Word Error Rate (WER) is one of the most widely used metrics for assessing ASR systems. It is calculated as the number of word-level insertions, deletions, and substitutions required to transform the system's output into the reference transcription, divided by the total number of words in the reference. A high WER implies significant errors in recognition, while a lower WER indicates that the transcribed text is closely aligned with the actual spoken content. WER is particularly useful for understanding how well the system captures the semantic structure of the input speech at the word level, which is critical for downstream tasks like summarization.

5.1.2 Character Error Rate (CER)

Character Error Rate (CER) serves a similar function to WER but evaluates errors at the character level. This is especially important for Indian languages, which often have complex morphology and character-based scripts. Even small character-level changes can alter the meaning of a word significantly. CER provides a finer-grained analysis of transcription errors, helping us understand the phonetic and orthographic fidelity of the ASR model, particularly in low-resource or script-heavy languages.

5.1.3 Recall-Oriented Understudy for Gisting Evaluation

ROUGE (Recall-Oriented Understudy for Gisting Evaluation) is a set of metrics used to evaluate the content quality of automatically generated summaries by comparing them against human-written reference summaries. In our evaluation, we use multiple variants of ROUGE: ROUGE-1 measures unigram (single word) overlap and captures basic content relevance. ROUGE-2 considers bigram (two-word) overlap, reflecting fluency and local coherence.ROUGE-L computes the longest common subsequence between the generated and reference summaries, indicating the degree to which important phrases are preserved. ROUGE-Lsum is optimized for evaluating summarization tasks with full sentence generation. These ROUGE metrics are especially important in abstractive summarization settings, where the model is expected to rephrase and condense content rather than copy it verbatim. High ROUGE scores suggest that the model captures the core meaning of the input while maintaining grammatical fluency and relevance.

5.2. Experiments

In our experiments, fine-tuning the Whisper ASR model on the multilingual speech corpus yielded dramatic gains in transcription accuracy. While the out-of-the-box Whisper baseline incurred a word error rate (WER = 1.051) and character error rate of 64.8%, adapting the model for two epochs on eight Indian languages reduced those errors to WER = 0.2676 and CER = 0.1266. The training loss curve shows a rapid initial drop followed by gradual smoothing, indicating that the model quickly learned language-specific acoustics without overfitting. These results underscore the importance of targeted multilingual data for robust ASR performance in underrepresented languages. Turning to summarization, the pretrained multilingual MT5 model proved unable to generate meaningful condensations of our news-style articles, achieving very low ROUGE scores (ROUGE-1 = 1.73%, ROUGE-2 = 0.24%, ROUGE-L = 1.61%, ROUGE-Lsum = 1.62%). After ten epochs of fine-tuning on concatenated XLSum data across the same eight languages, ROUGE-1 rose to 14.53%, ROUGE-2 to 3.67%, ROUGE-L to 12.02%, and ROUGE-Lsum to 12.12%. The summarization loss steadily declined throughout training, confirming stable convergence. The four- to eight-fold improvements in all ROUGE metrics demonstrate that domain- and language-specific fine-tuning is critical for generating coherent, concise summaries in a truly multilingual setting. Although quantitative evaluation of speech synthesis remains challenging,

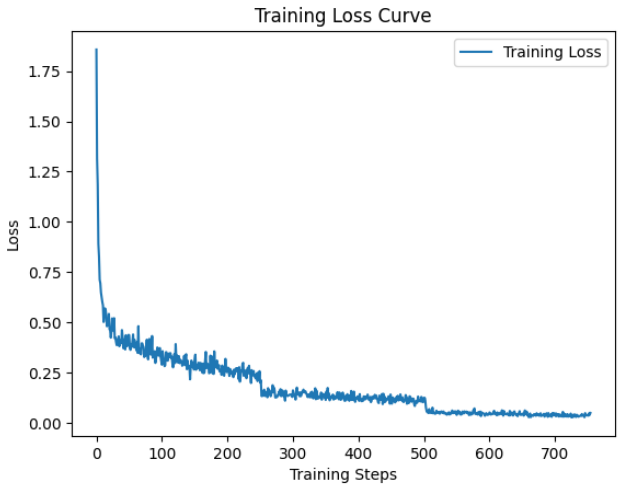

Figure 2. Training Loss curve on Fine Tuning Whisper-small model

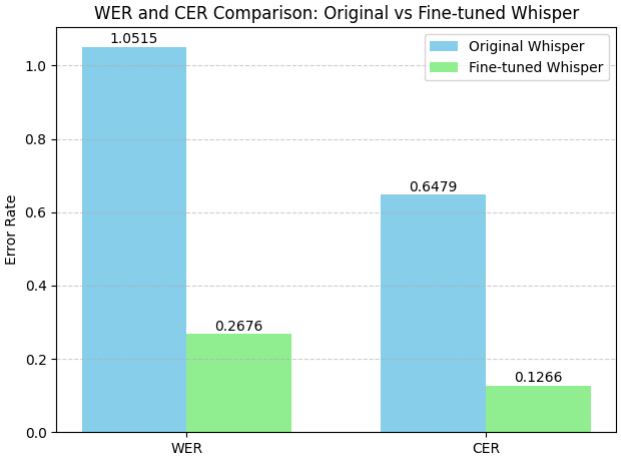

Figure 3. Comparison of Word Error Rate and Character Error Rate between finetuned and pretrained whisper model

informal listening tests of the Indic Parler-TTS system indicate that it produces highly intelligible, natural-sounding audio for both full transcriptions and their summaries. The model preserved appropriate prosody and phonetic characteristics for each language, maintained speaker consistency across concatenated chunks, and introduced no noticeable artifacts. Listeners found the synthesized speech clear and expressive, confirming that the TTS component effectively realizes the transition from text back to spoken output. Taken together, these results validate our end-to-end speech-to-speech summarization pipeline. By sequentially applying a fine-tuned ASR model, a fine-tuned multilingual summarizer, and a robust TTS system, we can transform raw audio in eight diverse languages into concise spoken summaries with high fidelity. The substantial improvements at each stage, coupled with natural synthesis quality, demonstrate both the feasibility and promise of multilingual speech summarization. This pipeline lays the groundwork for extending coverage to additional languages and domains, bringing accessible, language-agnostic speech tools to a broader audience.

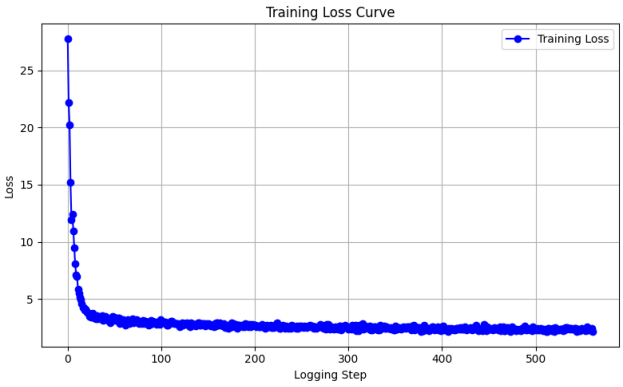

Figure 4. Training Loss Curve on Fine Tuning XL-Sum model

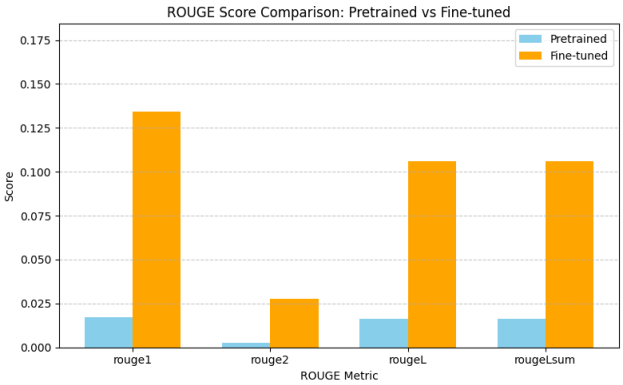

Figure 5. Comparison of ROGUE Scores before and after Fine tuning Xl-Sum Model

6. Conclusions

This work presents a complete, multilingual pipeline that transforms raw audio into natural-sounding speech summaries using a combination of automatic speech recognition, text summarization, language-aware text-to-speech synthesis, and synthetic data generation. By fine-tuning the Whisper model, we achieved a significant reduction in both Word Error Rate (WER) and Character Error Rate (CER), demonstrating improved transcription quality. Similarly, fine-tuning the mT5 summarization model led to marked improvements in ROUGE scores, highlighting its ability to produce more coherent and relevant summaries compared to the baseline.

Additionally, the integration of language detection and

automatic speaker assignment in the TTS module ensures that synthesized speech maintains linguistic and speaker consistency, enhancing listener experience. Our synthetic data generation approach, driven by the Indic Parler-TTS model and the XLSum dataset, helped mitigate dataset scarcity by producing high-quality audio-text pairs across several Indian languages.

Overall, the system exhibits robust performance across the full speech-to-speech pipeline, making it a promising framework for scalable, multilingual audio summarization applications. Future work may explore broader language support, cross-domain generalization, and real-time deployment capabilities.

References

[1] AI4Bharat. Indic parler-tts: Open-source text-to-speech for indian languages. https://huggingface.co/ai4bharat/indic-parler-tts, 2023. 2

[2] Yu Bai, Yang Gao, and Heyan Huang. Cross-lingual abstractive summarization with limited parallel resources. *arXiv preprint arXiv:2105.13648*, 2021. 2

[3] Alexis Conneau, Min Ma, Simran Khanuja, Yu Zhang, Vera Axelrod, Siddharth Dalmia, Jason Riesa, Clara Rivera, and Ankur Bapna. Fleurs: Few-shot learning evaluation of universal representations of speech. In *2022 IEEE Spoken Language Technology Workshop (SLT)*, pages 798–805. IEEE, 2023. 3

[4] Tahmid Hasan, Abhik Bhattacharjee, Md Saiful Islam, Kazi Samin, Yuan-Fang Li, Yong-Bin Kang, M Sohel Rahman, and Rifat Shahriyar. Xl-sum: Large-scale multilingual abstractive summarization for 44 languages. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4693–4703. Association for Computational Linguistics, 2021. 2, 3

[5] Kishore Kumar Mamidala and Suresh Kumar Sanampudi. Text summarization for indian languages: a survey. *Int J Adv Res Eng Technol (IJARET)*, 12(1):530–538, 2021. 1

[6] Alec Radford, Jong Wook Kim, Christopher Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Robust speech recognition via large-scale weak supervision. *arXiv preprint arXiv:2212.04356*, 2022. 2

[7] Fabian Retkowski, Maike Züfle, Andreas Sudmann, Dinah Pfau, Jan Niehues, and Alexander Waibel. From speech to summary: A comprehensive survey of speech summarization. *arXiv preprint arXiv:2504.08024*, 2025. 1

[8] Dhaval Taunk and Vasudeva Varma. Summarizing indian languages using multilingual transformers based models. *arXiv preprint arXiv:2303.16657*, 2023. 2

[9] Ashok Urlana, Pinzhen Chen, Zheng Zhao, Shay B. Cohen, Manish Shrivastava, and Barry Haddow. Pmindiasum: Multilingual and cross-lingual headline summarization for languages in india. *arXiv preprint arXiv:2305.08828*, 2023. 2

[10] Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 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*, pages 483–498, 2021. 2