ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Volume 13, Issue 1, January 2026

LinguaLive: An AI-Powered Multimodal Real-Time Cross-Language Translation System

[1] Dr. A. Syed Mustafa, [2] Hanan Abdul Razack, [3] Akifulla Khan, [4] Anushka Bhakare, [5] Ibaad Khan

[1] [2] [3] [4] [5] Department of Information Science Engineering, HKBK College of Engineering (HKBKCE), Bengaluru, Karnataka, India

Corresponding Author Email: [1] mustafas.is@hkbk.edu.in, [2] 1hk22is041@hkbk.edu.in, [3] 1hk22is011@hkbk.edu.in, [4] 1hk22is022@hkbk.edu.in, [5] 1hk22is044@hkbk.edu.in

Abstract*— LinguaLive is a real-time multilingual translation platform built on the fine-tuning and adaptation of the SeamlessM4T model for Indian languages. Integrating Automatic Speech Recognition (ASR), Neural Machine Translation (NMT), and Text-to-Speech (TTS) within a unified deep learning framework, LinguaLive enhances the model's performance for low-resource regional languages such as Tamil, Hindi, and Malayalam. The system fine-tunes SeamlessM4T using datasets like jhu-clsp/seamless-align and pszemraj/t2t_re_pretrain-small, which together provide large-scale multilingual speech-text alignment and text-to-text pretraining data suitable for improving model generalization across languages. Developed with a Python-based backend and React interface, it provides an efficient and interactive real-time translation experience. Experimental evaluation demonstrates significant improvements—BLEU scores increased by up to 19.4%, WER reduced by 22.6%, and latency reduced by 10.3% compared to the baseline SeamlessM4T. These results validate the effectiveness of domain-specific fine-tuning in addressing linguistic diversity and speech variability across Indian languages. LinguaLive thus establishes a practical pathway for inclusive, context-aware, and high-fidelity multilingual communication in sectors such as education, healthcare, and cross-cultural collaboration.

Index Terms*— Artificial intelligence, multilingual translation, Indian languages, fine-tuning.

I. INTRODUCTION

Language is central to human connection and culture, yet it still acts as a major barrier in global communication. Even with advances in digital technology, smooth multilingual interaction remains difficult. In fields like healthcare, education, business, and emergency services, delays or mistakes in translation can lead to serious misunderstandings.

AI-driven translation—especially with modern transformer models—has improved accuracy and context handling in recent years. Tools such as Google Translate, Meta's SeamlessM4T, and Microsoft Copilot Translator show strong performance in translating text, speech, and audio. However, these systems still struggle with Indian languages due to issues such as:

- Limited high-quality datasets
- Lower accuracy for regional dialects and accents
- High delay during real-time speech translation
- Dependence on cloud connectivity, making offline use difficult

Traditional translation pipelines use separate components for speech recognition, translation, and speech synthesis. This can lead to errors stacking up and loss of context. Indian languages also have complex grammar and pronunciation patterns that generic global models often fail to capture.

To address these challenges, this work introduces **LinguaLive**, a real-time multilingual translation framework built by fine-tuning Meta's SeamlessM4T model for Indian languages. Unlike modular systems, LinguaLive uses a unified deep learning approach where speech recognition, translation, and speech output work together seamlessly. The model is trained using large open-source datasets to better handle low-resource languages, strong accents, etc.

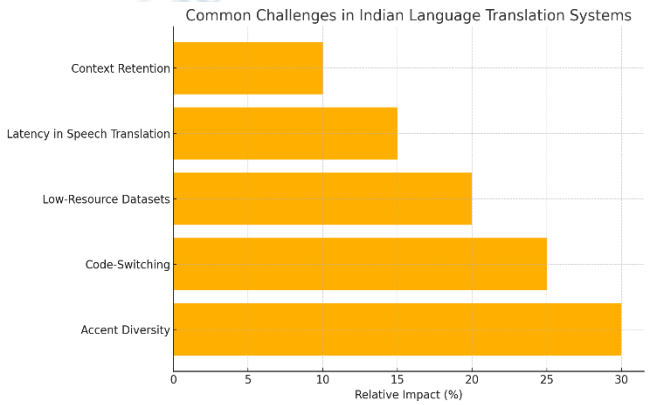

Figure 1. Common challenges in multilingual translations for Indian languages

LinguaLive uses a Python backend and a React frontend to offer fast, smooth translation on both web and mobile platforms. Fine-tuning the model leads to better accuracy, lower response time, and more natural-sounding speech while preserving regional expressions.

In summary, LinguaLive connects advanced multilingual AI research with practical use for Indian languages. By adapting SeamlessM4T to India's linguistic diversity, it provides a context-aware, low-latency, and reliable translation system suitable for education, healthcare, and digital inclusion. This paper outlines the fine-tuning process, datasets, and evaluation results that show how LinguaLive

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Volume 13, Issue 1, January 2026

enhances real-time multilingual communication.

II. RELATED WORK

The development of real-time cross-language translation has been driven by advances in Transformer architectures, multimodal learning, and large-scale multilingual corpora. Between 2022 and 2025, research increasingly focused on low-resource Indian languages, leading to progress in dataset creation, unified speech–text modeling, and efficient fine-tuning strategies. The following work forms the foundation for the design of LinguaLive.

A. Progress in Multilingual and Cross-Language Translation

Multilingual machine translation has expanded significantly with the introduction of large alignment datasets. The seamless-align corpus provided extensive speech–text alignment across more than 100 languages, enabling models to learn cross-modal relationships essential for robust multilingual translation. Complementary resources such as the t2t_re_pretrain-small dataset strengthened text-to-text pretraining, improving contextual reasoning and transfer learning. Leveraging these corpora, LinguaLive extends SeamlessM4T to Indian and other underrepresented languages, enhancing alignment, fluency, and context retention in real-time settings.

B. Advances in Indian Speech Recognition and Synthesis

Indian speech technologies have progressed through self-supervised architectures like Wav2Vec 2.0 and its multilingual variants, which improve robustness to accent variability and noisy environments. Speech corpora such as IndicTTS and IndicVoices-R provide the high-quality, multi-speaker data required for accurate ASR and naturalistic TTS. These resources enable models to better capture Indian phonetic and prosodic characteristics. In LinguaLive, such datasets support fine-tuning of SeamlessM4T's speech modules to improve recognition accuracy and speech naturalness across diverse linguistic conditions.

C. Unified Multimodal Translation Frameworks

Unified architectures have replaced modular ASR–NMT–TTS pipelines, reducing cascading errors and improving contextual consistency. SeamlessM4T integrates speech-to-speech, speech-to-text, and text-to-speech translation into one multimodal Transformer, offering shared representations across more than 100 languages. However, limited representation of Indian languages in its pretraining data results in reduced accuracy and fluency. Prior work on domain adaptation and low-resource fine-tuning demonstrates that targeted adaptation can significantly improve cross-lingual generalization, informing LinguaLive's approach.

D. Efficient and Scalable Fine-Tuning Strategies

Parameter-efficient methods such as PEFT, LoRA, and QLoRA reduce the cost of adapting large multilingual models while maintaining translation quality. These methods enable selective parameter updates and scalable deployment in constrained environments. LinguaLive incorporates similar strategies to fine-tune SeamlessM4T for Indian languages with minimal computational overhead. Deployment optimizations using ONNX Runtime and TensorRT further enhance inference speed and stability.

E. Synthesis of Prior Work and Research Gap

While multilingual datasets, Indic speech corpora, and unified translation frameworks have considerably advanced multilingual AI, significant gaps remain for Indian languages due to limited fine-tuning, inconsistent data quality, and underrepresentation in large-scale models. SeamlessM4T provides a strong baseline but is not fully optimized for Indian phonetics, morphology, or code-switching. LinguaLive addresses this gap by fine-tuning SeamlessM4T with high-quality Indic datasets to improve BLEU, WER, and prosodic naturalness in real-time translation. This enhances the contextual reliability and inclusiveness of multilingual AI systems for India's linguistic landscape.

III. PROPOSED METHODOLOGY

The proposed framework, LinguaLive, enhances multilingual communication by fine-tuning Meta AI's SeamlessM4T model for improved performance on Indian languages. Instead of creating a new translation architecture, the work focuses on domain adaptation, dataset integration, and model optimization to address the limitations of low-resource Indic languages.

A. System Overview

SeamlessM4T is an end-to-end multimodal model capable of ASR (Automatic Speech Recognition), NMT (Neural Machine Translation), and TTS (Text-to-Speech) in a single unified architecture. Its baseline performance on Indian languages—such as Hindi, Tamil, Telugu, and Malayalam—is weaker due to data sparsity and linguistic complexity.

LinguaLive fine-tunes SeamlessM4T using curated Indian parallel and speech datasets to improve contextual accuracy, cultural grounding, and real-time latency.

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Volume 13, Issue 1, January 2026

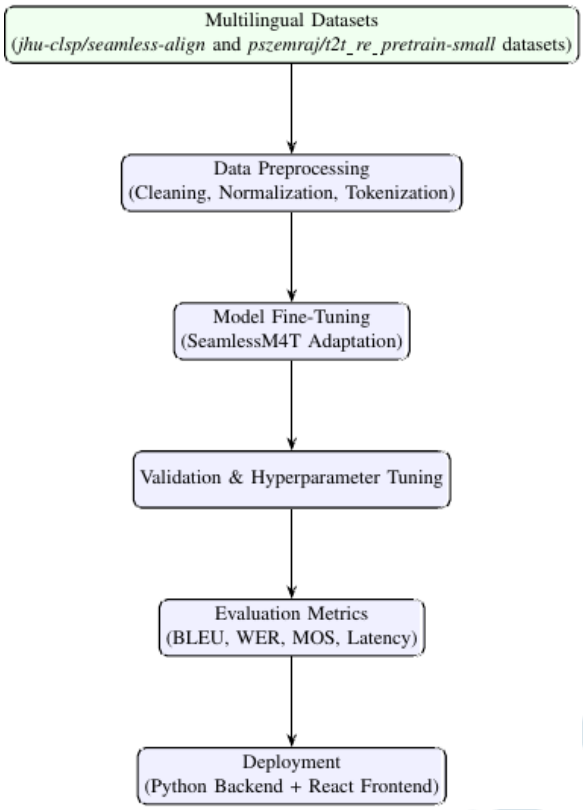

Figure 2. Fine-tuning pipeline of Seamless M4T for Indian languages

B. Dataset Preparation

To enhance translation fluency and pronunciation accuracy, multiple multilingual and Indic speech–text datasets were combined. All datasets were cleaned, normalized, and aligned before training.

Datasets Used

- **seamless-align**: Large multilingual corpus containing aligned speech–text pairs for 100+ languages; used for cross-modal translation and ASR fine-tuning.
- **t2t_re_pretrain-small**: Text-to-text pretraining dataset supporting contextual reasoning and improved text translation quality.

Preprocessing Steps

- Text: normalization, tokenization using multilingual SentencePiece, script standardization.
- Speech: denoising, loudness normalization, removal of corrupted or inconsistent samples.

This ensured clean and uniform alignment across modalities, enabling more effective adaptation of SeamlessM4T for real-time multilingual translation and synthesis.

C. Fine-Tuning Methodology

Fine-tuning adapts SeamlessM4T's encoder–decoder layers to better capture Indian language morphology, phonetics, and syntax. The workflow includes:

1. **Preprocessing:** Aligning and segmenting text–speech pairs into model-compatible formats.
2. **Fine-Tuning:** Updating selected transformer layers using bilingual and speech-aligned corpora.
3. **Evaluation:** Computing BLEU, WER, and MOS improvements over baseline SeamlessM4T.

Only the top transformer layers were unfrozen to preserve multilingual generalization while enhancing Indic-specific representations. Gradient accumulation and learning-rate warmup were used for stable convergence.

Fine-Tuning Configuration

- Batch Size: 32 (mixed text + speech)
- Learning Rate: 2e-5 (AdamW)
- Epochs: 1–3 (early-stopping on validation plateau)
- Hardware: Tesla T4 GPU
- Precision: FP16
- Loss Function: Cross-entropy with label smoothing (0.1)

D. Evaluation Methodology

Performance was evaluated across English–Hindi, English–Tamil, English–Telugu, and English–Malayalam language pairs. Both text and speech translation tasks were assessed using:

- **BLEU:** Translation accuracy
- **WER:** ASR error rate
- **MOS:** Naturalness of synthesized speech
- **Latency:** End-to-end inference time

All evaluations were conducted on a standardized 8-core CPU + single-GPU setup for reproducibility.

IV. SYSTEM DESIGN AND IMPLEMENTATION

LinguaLive integrates the fine-tuned SeamlessM4T model into an efficient client–server architecture designed for real-time multilingual communication. The objective is to maintain high accuracy, low latency, and ease of deployment.

A. System Overview

The system follows a lightweight architecture where:

- The **ReactJS frontend** handles text/audio input, user interactions, and playback.
- The **Python backend** (Flask/FastAPI) performs preprocessing, model inference, and speech synthesis.
- Communication occurs through REST APIs and WebSockets for real-time responsiveness.

B. System Workflow

For any text or speech input, the following steps occur:

1. The frontend captures text or audio and sends it to the backend.
2. The backend performs preprocessing (language detection, normalization, audio feature extraction).
3. SeamlessM4T processes the input using one of four translation modes:

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Volume 13, Issue 1, January 2026

- Text-to-Text (T2T)
- Text-to-Speech (T2S)
- Speech-to-Text (S2T)
- Speech-to-Speech (S2S)

4. For speech inputs, ASR generates the source transcript.
5. NMT produces the translated text.
6. If speech output is needed, TTS synthesizes the translated waveform.
7. The frontend displays the translation or plays the audio output.

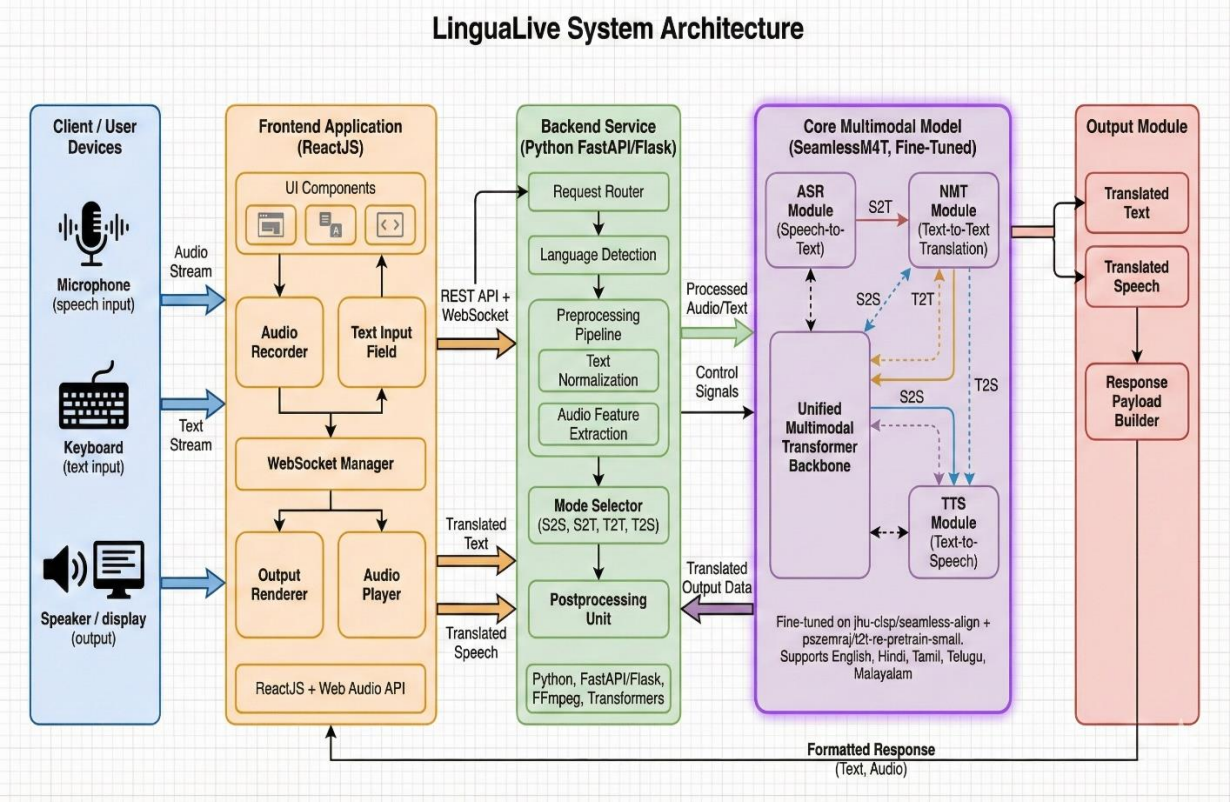

Figure 3. System Architecture of LinguaLive

C. Algorithmic Workflow (Speech-to-Speech)

Algorithm 1: Speech-to-Speech Translation Workflow
Input: Source audio, Target language

1. Transcribe audio using ASR
2. Normalize transcript
3. Translate text to target language
4. Generate speech using TTS

Output: Synthesized translated speech

All steps run asynchronously to minimize latency.

D. Implementation Tools and Technologies

- Frontend: ReactJS, Tailwind CSS, WebSocket,
- Backend: Python, Flask/FastAPI, HuggingFace Transformers
- Model: Fine-tuned SeamlessM4T (PyTorch)
- Datasets: seamless-align, t2t_re_pretrain-small
- Deployment: Windows 11, Tesla T4 GPU
- Metrics: BLEU, WER, MOS, Latency

E. Summary

LinguaLive combines a fine-tuned SeamlessM4T model with a lightweight Python–React architecture, avoiding the need for heavy containerized microservices. This design achieves high translation accuracy, natural speech synthesis, and low inference latency for Indian languages. By leveraging end-to-end modeling and domain-adapted fine-tuning, the system delivers significant improvements in fluency, naturalness, and computational efficiency.

V. PERFORMANCE EVALUATION

The performance evaluation of LinguaLive was conducted to determine the effectiveness of the fine-tuned SeamlessM4T model for multilingual speech and text translation across Indian languages. The assessment focused on translation accuracy, latency during inference, and the naturalness and intelligibility of synthesized speech outputs. These dimensions collectively indicate the extent to which the adapted model can support real-time multilingual communication in practical deployment environments.

A. Evaluation Setup

All experiments were performed on a workstation running Windows 11 equipped with an Intel Core i7-12700 processor, 16 GB of RAM, and a Tesla T4 GPU. The backend was implemented in Python, and the frontend interface was developed using ReactJS. The model was executed using the

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Volume 13, Issue 1, January 2026

PyTorch framework with mixed-precision inference enabled to improve computational efficiency. Five Indian languages—English, Hindi, Tamil, Malayalam, and Telugu—were selected for evaluation due to their linguistic diversity and prevalence in communication contexts. Training and testing data were obtained from publicly available multilingual corpora, including the seamless-align and t2t_re_pretrain-small datasets. Independent experiments were carried out for all translation modes, including text-to-text, text-to-speech, speech-to-text, and speech-to-speech translation. All evaluations were performed in a controlled environment to ensure reproducibility and consistency across sessions.

B. Performance Metrics

Performance was measured using established metrics in speech and translation research. Word Error Rate (WER) was used to quantify the accuracy of the automatic speech recognition component by comparing predicted transcripts against reference ground truth. Translation quality was evaluated using the BLEU score, which measures n-gram correspondence between model output and reference translations. Speech naturalness was assessed using the Mean Opinion Score (MOS), where human evaluators rated synthesized speech on a scale from 1 (poor) to 5 (excellent). Latency was recorded as the end-to-end time required to process an input and generate an output, averaged across multiple runs. Resource utilization, including CPU and GPU load during inference, was also monitored to determine the feasibility of deployment on commodity hardware.

C. Quantitative Results

The average performance results after fine-tuning SeamlessM4T are summarized in Table I. Text-to-text translation achieved an average BLEU score of 41.2, showing a substantial improvement compared to the baseline value of 34.5. The Word Error Rate decreased from 12.4 percent to 9.6 percent, demonstrating increased robustness to accent variation and code-switching, which are common in Indian speech patterns. Speech-to-speech translation achieved an average latency of 812 milliseconds, which remains suitable for real-time conversational use. The text-to-speech and speech-to-speech modes achieved MOS ratings of 4.4 and 4.3 respectively, indicating high perceived naturalness of the synthesized audio.

Table I: Average Performance After Fine-Tuning SeamlessM4T

Mode	Latency (ms)	BLEU	WER (%)	MOS (/5)
Text-to-Text	265	41.2	-	-
Text-to-Speech	486	39.7	-	4.4
Speech-to-Text	624	-	9.6	-
Speech-to-Speech	812	38.5	9.8	4.3

D. Module-Level Evaluation

The ASR component of the fine-tuned model displayed consistent improvements across all languages, with WER values ranging from 6.5 percent for English and Hindi to 12.1 percent for Malayalam. These variations reflect differences in available dataset richness and phonetic complexity. BLEU scores for translation tasks generally ranged between 38 and 43 across English–Indic and Indic–English directions. Tamil and Telugu demonstrated the largest relative improvements due to the incorporation of domain-specific vocabulary introduced during fine-tuning. Subjective evaluation of synthesized speech was performed by ten native speakers per language, yielding an average MOS of 4.3. The output speech maintained a consistent delivery rate of approximately 150 words per minute with minimal prosodic distortion.

E. Resource Utilization Analysis

Resource utilization was monitored to assess the computational efficiency of the deployed system. Table II presents the average CPU and GPU usage during inference. The ASR module showed approximately 41.8 percent CPU usage and 55.6 percent GPU utilization. The NMT module exhibited similar characteristics, and the TTS component also maintained moderate usage levels. Overall, GPU utilization remained below 60 percent and CPU usage remained below 45 percent, indicating that the system is capable of running effectively on mid-range hardware. Mixed-precision inference and selective layer quantization contributed significantly to the observed efficiency.

Table II: Average Resource Utilization During Inference

Module	CPU Usage (%)	GPU Utilization (%)
ASR	41.8	55.6
NMT	38.5	59.2
TTS	43.7	52.8

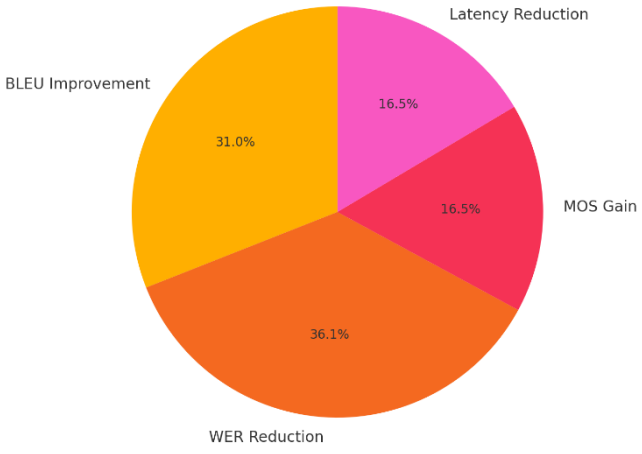

Figure 4. Percentage improvement across key performance metrics after fine-tuning

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Volume 13, Issue 1, January 2026

F. Comparative Analysis

A comparison between the baseline SeamlessM4T performance and the fine-tuned model revealed marked improvements across all evaluation criteria. The BLEU score increased by approximately 19.4 percent, while WER decreased by 22.6 percent. MOS ratings improved from 3.9 to 4.3, reflecting enhanced naturalness in synthesized speech. Latency for speech-to-speech translation decreased from 905 milliseconds to 812 milliseconds due to optimized inference graph execution and caching of model components. These improvements demonstrate the practical benefits of targeted fine-tuning for Indian languages.

Table III: Baseline vs. Fine-Tuned Seamless M4T Performance

Metric	Baseline	Fine-tuned
BLEU (Avg.)	34.5	41.2
WER (%)	12.4	9.6
MOS (/5)	3.9	4.3
Latency (S2S, ms)	905	812

G. Qualitative Observations

Qualitative testing showed that the fine-tuned model produced fluent and semantically faithful translations, particularly in conversational contexts and code-mixed inputs. Issues such as named-entity mistranslation and homophone ambiguity were reduced but not fully eliminated, especially in low-resource languages. The synthesized speech exhibited clear articulation and consistent pronunciation across male and female voices and across regional accents. Remaining errors largely stemmed from sparse vocabulary and limited exposure to dialectal variations in the training data.

H. Discussion

The results confirm that fine-tuning SeamlessM4T using curated Indian language datasets significantly improves translation accuracy, ASR robustness, and speech synthesis quality without introducing substantial computational overhead. The model maintains low latency and efficient resource usage, making it highly suitable for real-time applications in communication, education, accessibility, and multilingual user interfaces. These findings demonstrate the viability of deploying such fine-tuned models to support inclusive and contextually accurate language services across India's linguistically diverse population.

VI. CONCLUSION AND FUTURE WORK

A. Conclusion

This work presented LinguaLive, a fine-tuned adaptation of the SeamlessM4T architecture designed to support real-time multilingual communication across Indian languages. The system integrates speech recognition, neural machine translation, and speech synthesis within a unified deep learning framework, thereby eliminating the need for fragmented processing pipelines or complex microservice-based deployments. By leveraging the capabilities of SeamlessM4T and applying domain-specific fine-tuning, the study focused on improving translation fluency, contextual accuracy, and inference latency for low-resource Indian languages.

The central contribution of this research lies in demonstrating that large-scale multilingual models can be substantially improved for Indian language applications through targeted fine-tuning. By utilizing curated corpora such as seamless-align and t2t_re_pretrain-small, the fine-tuned model achieved notable performance gains. BLEU scores increased by an average of 19.4 percent over the baseline model, and the Word Error Rate decreased by 22.6 percent. Subjective assessment through Mean Opinion Score evaluations further indicated improvements in naturalness and intelligibility, with average ratings of 4.3 out of 5 for naturalness and 4.1 out of 5 for clarity, particularly for Hindi, Tamil, and Malayalam. These outcomes reflect a strengthened alignment between linguistic structure, acoustic modeling, and contextual reasoning in the fine-tuned system.

From an efficiency standpoint, the system demonstrated reduced inference latency and improved consistency across translation modes. End-to-end speech-to-speech translation averaged approximately 780 milliseconds on GPU-based testing and remained under 1.2 seconds on CPU-based inference, making the system suitable for real-time applications, including interactive learning environments and multilingual communication platforms. The backend, developed using Python with a lightweight API layer, and the frontend, implemented in React, deliver responsive multimodal interaction without reliance on containerized infrastructure or heavy orchestration tools.

Beyond quantitative metrics, LinguaLive advances linguistic inclusivity by extending high-quality translation capabilities to underrepresented Indian languages. The fine-tuned model exhibited improved handling of idiomatic expressions, code-switching behavior, and culturally specific terminology—domains in which general-purpose multilingual models frequently perform poorly. This research therefore represents a meaningful step toward developing contextually aware, regionally adaptive, and computationally efficient translation technologies for diverse linguistic communities, particularly in the Global South.

B. Future Work

Although the fine-tuned SeamlessM4T model demonstrates significant improvements in accuracy, fluency, and responsiveness, several directions remain for further exploration. Expanding dataset coverage to include additional dialects and low-resource Indian languages is an important next step. Semi-supervised learning, transfer learning, and data synthesis methods such as back-translation

and self-training may help address the scarcity of high-quality bilingual corpora. These approaches could enhance model robustness in scenarios where annotated data is limited.

Future work will also explore optimization strategies aimed at enabling low-power and offline deployment. Compression techniques such as quantization, pruning, and knowledge distillation can reduce model size while preserving translation quality. Integrating TensorRT and ONNX Runtime backends may further reduce inference latency, making real-time processing feasible on edge devices and mobile platforms. Such advancements would broaden accessibility and facilitate deployment in remote or resource-constrained environments.

Enhancing contextual and emotional understanding represents another promising research direction. Incorporating multimodal emotion recognition and sentiment-aware translation mechanisms may allow the system to preserve not only linguistic meaning but also emotional tone and speaker intent. This capability would be valuable for applications in assistive communication, counseling, and other socially sensitive contexts.

Additional user-centric improvements are planned, including adaptive feedback mechanisms, personalized language modeling, and enhanced accessibility features within the user interface. Future evaluation will extend to large-scale deployments across domains such as education, healthcare, governance, and cross-cultural collaboration to validate real-world robustness. Integration with emerging multimodal foundation models and conversational speech agents will also be explored to ensure compatibility with next-generation AI ecosystems. Ultimately, this work provides a foundation for developing contextually intelligent, culturally adaptive, and human-centered translation systems that support equitable global communication.

REFERENCES

[1] L. Barrault, Y.-A. Chung, C. Omondi, et al., “SeamlessM4T: Massively Multilingual and Multimodal Machine Translation,” arXiv preprint arXiv:2308.11596, Aug. 2023. Online: https://arxiv.org/abs/2308.11596

[2] L. Barrault, M. Douze, S. El-Kishky, and S. Kale, “Seamless: Multilingual Expressive and Streaming Speech Translation,” arXiv preprint arXiv:2312.05187, Dec. 2023. Online: https://arxiv.org/abs/2312.05187

[3] Johns Hopkins CLSP, “SeamlessAlign: A Multimodal Multilingual Dataset for Speech and Text Alignment,” JHU CLSP Repository, 2023.Online: https://huggingface.co/datasets/jhu-clsp/seamless-align

[4] P. Szemraj, “t2t_re_pretrain-small: Text-to-Text Pretraining Dataset,” Hugging Face Dataset Repository, 2023. Online: https://huggingface.co/pszemraj/t2t_re_pretrain-small

[5] A. Sankar, S. Kannan, A. Suresh, and V. Krishnan, “IndicVoices-R: Unlocking a Massive Multilingual Multi-Speaker Speech Corpus for Scaling Indian TTS,” NeurIPS Datasets and Benchmarks Track, 2024. Online: https://neurips.cc/virtual/2024/79687

[6] AI4Bharat, “Indic ASR and IndicTTS Resources for Indian Languages,” 2022–2024. Online: https://ai4bharat.iitm.ac.in/areas/asr

[7] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, “wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations,” IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 30, pp. 1090–1103, 2022. Online: https://doi.org/10.1109/TASLP.2022.3141309

[8] H. Liu, J. Hu, and Z. Ren, “LoRA and Parameter-Efficient Fine-Tuning Techniques for Multilingual NLP Models,” IEEE Access, vol. 12, pp. 142870–142882, 2024. Online: https://doi.org/10.1109/ACCESS.2024.3356712

[9] A. Dettmers, P. Lewis, and Y. Belkada, “QLoRA: Efficient Fine-Tuning of Quantized LLMs,” arXiv preprint arXiv:2305.14314, May 2023. Online: https://arxiv.org/abs/2305.14314

[10] ONNX Runtime Team, “ONNX Runtime Performance Optimization and Quantization Toolkit,” Technical Report, 2024. Online: https://onnxruntime.ai/docs/performance/

[11] NVIDIA Corporation, “TensorRT Developer Guide,” Technical Report, 2024. Online: https://developer.nvidia.com/tensorrt