Real-Time Multilingual Speech Translation for Peer Communication

B Meenakshi[1], Mohammed Wajahat Hussain[2], Mittapalli Arvind Sai[3]

[1,2,3]Department of Information Technology, Mahatma Gandhi Institute of Technology, Gandipet, Hyderabad, Telangana, India.

***Emails:** bmeenakshi_it@mgit.ac.in[1], mwajahat_csb213243@mgit.ac.in[2], marvind_csb213242@mgit.ac.in[3]*

Abstract

Language continues to be a major obstacle to effective communication in a world that is becoming more interconnected by the day. This paper presented a real-time audio translation system that facilitates multilingual communication during peer-to-peer video calls. The application enables natural communication in the user's preferred language by utilizing WebRTC for low-latency media transmission and incorporating sophisticated AI models such as Whisper for speech-to-text, GPT for language translation, and gTTS for text-to-speech synthesis. In addition to allowing real-time subtitle overlays and translated audio playback during conversations, the system supports five other languages: English, Hindi, Tamil, Telugu, and German. Low latency, scalability, and user-centric design are prioritized in the architecture, which is constructed with a Fast API backend and a React-based front-end. We address issues such as translation delays, synchronization, and audio buffering, and assess the system using user experience, latency benchmarks, and qualitative performance.

***Keywords:** WebRTC, Real-Time Translation, Multilingual Communication, Speech-to-Speech Translation.*

1. Introduction

In an increasingly interconnected world, the demand for seamless real-time communication across language boundaries has increased significantly. Traditional approaches to language interpretation and translation, such as human interpreters or static subtitles, are not always feasible in dynamic real-time scenarios. This gap has led to the development of automated systems that are capable of performing live speech translation with minimal latency. The intersection of translation, [1-3] interpretation, and accessibility has become a central concern in designing inclusive communication platforms, particularly in domains such as education, media, and global conferencing The project employs the principles of Artificial Neural Networks to accurately map gestures to their respective meanings. By translating gestures into audible speech, the Digital Vocalizer facilitates real-time communication, breaking down the barriers that mute and deaf individuals face daily. Our project builds upon this body of work by developing a real-time translation application using WebRTC for low-latency media transmission and integrating state of-the-art AI components: whisper for speech-to-text, GPT for neural translation, and gTTS for text-to-speech synthesis. Unlike prior systems that focused on offline translation or subtitle-only output, our implementation supports bidirectional peer-to-peer communication with both live audio translation and caption overlay. This expands upon the ideas of Annamareddy et al. and Sangolgi et al. and applies them to conversational interfaces. The system also benefits from recent advancements in natural language processing pipelines and offers flexibility for multilingual real time interaction in a scalable and user-centric architecture. Our project builds upon this body of work by developing a real-time translation application using WebRTC for low-latency media transmission and integrating state of-the-art AI components: whisper for speech-to-text, GPT for neural translation, and gTTS for text-to-speech synthesis. Unlike prior systems that focused on offline translation or subtitle-only output, our implementation supports bidirectional peer-to-peer communication with both live audio translation and caption overlay. This expands upon the ideas of Annamareddy et al. and Sangolgi et al. and applies them to conversational interfaces. The system also

benefits from recent advancements in natural language processing pipelines and offers flexibility for multilingual real time interaction in a scalable and user-centric architecture. [4]

2. Methodologies

This system leverages WebRTC for real-time, low-latency, peer-to-peer video and audio communication between users. WebRTC enables direct media streaming between browsers without external dependencies, making it well suited for scalable and efficient conferencing systems. The frontend was developed using React and Next.js, offering an intuitive interface where users can initiate calls, select their preferred language, and view translated subtitles during the conversation. WebRTC's efficient handling of real time media exchanges makes it a strong foundation for live multilingual applications.

The backend architecture employs FastAPI as the server framework to coordinate the AI-powered translation tasks. The incoming audio streams are processed in real time using OpenAI's Whisper model, which performs multilingual speech-to-text transcription with high accuracy and robustness across accents and noise conditions . The transcribed text is passed to a GPT-based neural translation engine that translates the text into the recipient's preferred language using contextual awareness to improve semantic accuracy. The translated text is then either displayed as subtitles or sent through Google Text-to-Speech (gTTS) to generate a translated voice response, which is returned to the client over the WebRTC's data channel [5]

2.1.System Design

This system enables real-time multilingual communication using peer-to-peer video calls. It is built around WebRTC for media streaming and Fast API for backend processing. The frontend, built using React and Next.js, allows users to initiate a call, choose their preferred language, and see translated captions or hear translated audio in real time. Each peer connects directly to minimize latency and supports scalable multi-user communication.

2.2.Speech Processing Pipeline

When a user speaks, the captured audio is chunked and streamed to the backend, where whisper is used for multilingual speech-to-text transcription . The resulting text is then passed through a GPT-based model to perform language translation, considering the context and fluency . The translated text is either displayed as captions or converted into audio using gTTS and streamed back over the WebRTC data channel to the receiver.

2.3.Media and Caption Synchronization

To manage synchronization, the system uses timestamp-based chunking and buffer control to ensure that the translated audio and captions are aligned with the speaker's video stream. Captions are dynamically rendered onto a video container to simulate professional subtitling systems. A monitoring dashboard tracks the caption latency, audio processing delays, and user-selected language flows to evaluate the system health and performance in real time.

3. Results and Discussion

This system enables real-time multilingual communication with seamless caption generation and translated audio playback. Latency, translation accuracy, and user feedback were evaluated to assess the performance under various network and language 3 scenarios. The primary metrics observed included the caption delay, audio translation turnaround, and subjective intelligibility ratings across multiple language pairs. Tests were conducted on a controlled network to isolate system-induced delays from bandwidth- or jitter-related variations. The results showed that the architecture maintained a consistent user experience under controlled concurrency. The average memory usage per session remained within acceptable bounds, and the caption refresh times did not vary significantly with session duration or user speech rate. These results are consistent with prior evaluations of modular NLP systems designed for real-time communication. [6]

3.1.Translation Pipeline Visualization

The pipeline processes audio in 3–5 second chunks using Whisper for transcription, GPT for translation, and gTTS for speech synthesis. Each component is containerized for independent scalability to prevent bottlenecks when handling simultaneous user requests. Captions and translated audio are typically returned in under 1.5 seconds, depending on the language pair and the chunk length. These processing benchmarks are consistent with the latency-sensitive translation systems explored in recent multilingual

NLP research. (Figure 1) [7-8]

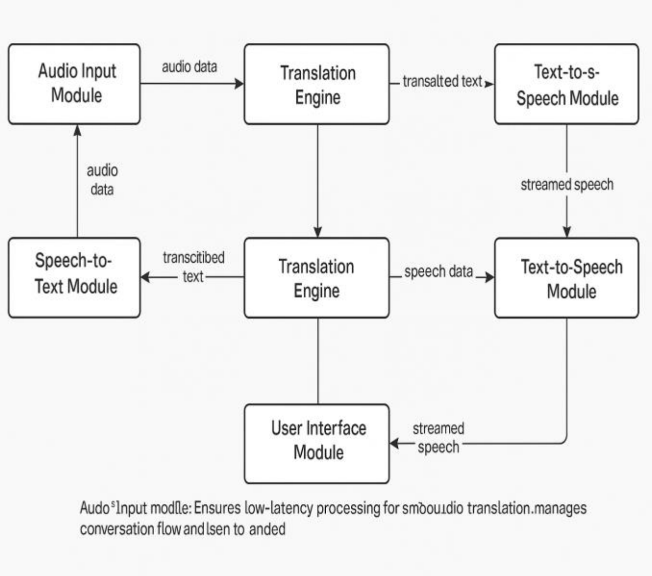
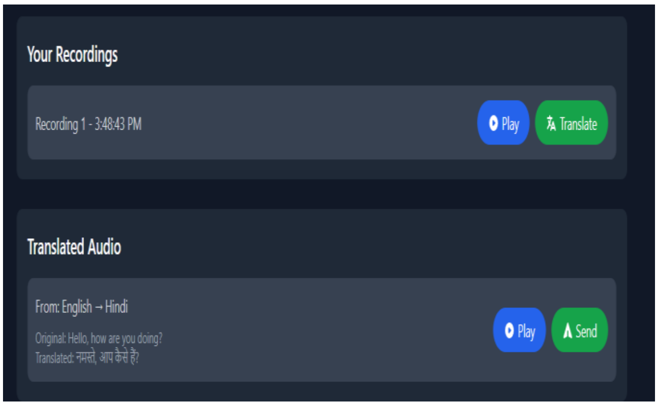


Figure 1 **Translation Pipeline Visualization**

Across all test sessions, the system maintained a transcription accuracy of greater than 93% for languages supported by Whisper's multilingual model. English–Hindi pairs showed high reliability in both transcription and translation, whereas longer 4 sentences in English–Telugu occasionally produced incomplete outputs. These issues are typically related to the token limits in streaming pipelines and can be mitigated by buffering enhancements [9-10]

3.2.Live Translation and Subtitles

The captioning system overlays the translated text directly onto the video stream in real-time. It uses time-stamped segments and dynamic DOM manipulation for responsive UI updates, which are essential for smooth communication during fast-paced exchanges. Subtitles are updated every 3–5 s and placed near the speaker video tile for visual alignment. The synchronization was evaluated under varying network conditions. During the medium-bandwidth tests (5–10 Mbps), the subtitles remained consistently aligned with the speaker's video. However, in low-bandwidth conditions (¡2 Mbps), we observed slight delays in caption updates, mostly owing to the extended audio upload or Whisper's transcription latency. The system dynamically adjusts the caption buffer size to compensate for these variations. (Figure 2)


Your Recordings

Recording 1 - 3:48:43 PM

Translated Audio

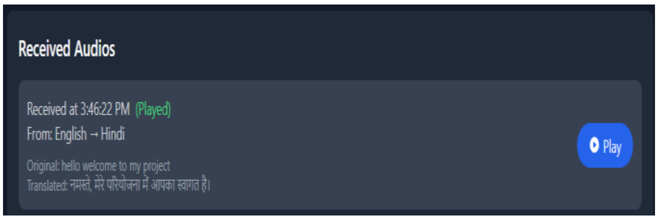
From: English → Hindi

Original: Hello, how are you doing?
Translated: नमस्ते, आप कैसे हैं?

Figure 2 **Live Translation and Subtitles**

3.3.Audio Translation Feedback

Audio translation performance was evaluated using gTTS, which synthesizes translated speech in real-time after receiving text from the GPT. On average, the translated audio was played back within 2.2 seconds from the moment a user finished speaking. 5 This delay included whisper transcription (0.8s), GPT translation (0.5s), and TTS synthesis and transmission (0.9s). The audio output was subjectively evaluated by bilingual testers and rated for intelligibility, tone quality, and timing. Languages with high phoneme overlap with English (e.g., German, Hindi) scored the highest, with clear pronunciation and tone modulation. Some minor artifacts were observed in the synthesized Telugu and Tamil outputs, primarily owing to limitations in open-source phoneme rendering and prosody modeling. Future improvements may include migrating to more advanced TTS engines, such as Tacotron or FastSpeech, or fine-tuning multilingual voice synthesis for underrepresented languages. Additionally, the caching of common phrases and parallel audio synthesis could further reduce the overall playback delay, aligning with the optimizations seen in streaming audio platforms. (Figure 3)


Received Audios

Received at 3:46:22 PM (Played)
From: English → Hindi

Original: hello welcome to my project
Translated: नमस्ते, मेरे परियोजना में आपका स्वागत है।

Figure 3 **Audio Translation Feedback**

3.4.Scalability and Language Handling

The system was designed with scalability and modularity at its core to accommodate growing user bases and expanding language support. Each core component—Whisper for transcription, GPT for translation, and gTTS for speech synthesis—operates as an independent service in the backend pipeline. This separation allows for horizontal scaling, where individual modules can be replicated or containerized to handle the increased loads across multiple sessions. To ensure the seamless addition of new languages, the architecture adopts a plug and-play approach. Support for additional whisper-compatible transcription models and gTTS voice profiles can be added through configuration updates, without altering the core logic . This flexibility makes the system adaptable for future expansions, such as Arabic, Spanish, or other low-resource languages, which are increasingly relevant in multilingual global deployments [11-14]

Future Scope

The real-time multilingual communication system developed in this project has shown promising capabilities, particularly in the context of peer-to-peer video calling. It successfully combines core components such as speech recognition, language translation, and speech synthesis to deliver cross-lingual interaction without noticeable latency. This setup lays a strong foundation for multilingual collaboration across various domains. Nevertheless, the system's potential can be further enhanced by adopting more efficient and lightweight components, especially for environments with limited computational power or unstable internet connections. Replacing heavy ASR models like Whisper with optimized, on-device alternatives could allow the system to function in offline or edge-based scenarios, making it more accessible and resilient in low-resource settings. To improve the user experience, especially in terms of voice output quality, it would be beneficial to upgrade the current speech synthesis module. Integrating neural TTS technologies such as Tacotron 2 or FastSpeech may lead to more expressive and human-like voice generation. These models offer smoother prosody, better pronunciation, and a more natural rhythm, which are particularly useful when dealing with languages that lack large-scale datasets. Implementing such models can enhance comprehension and engagement, especially when the system is used for educational, corporate, or assistive communication purposes. Furthermore, the system's adaptability can be increased by incorporating real-time speaker identification and automatic language detection. These additions would be instrumental in handling conversations involving multiple participants and varying languages. With speaker diarization, the system could assign subtitles or voice outputs to specific individuals, improving clarity in group interactions. Automatic detection of spoken language would allow for seamless switching between translation pipelines, eliminating the need for manual input and enabling a more fluid user experience. These features are crucial for large-scale deployments such as international meetings, multilingual classrooms, and global customer support platforms. Lastly, consistency in translated content remains a significant factor, especially during long or technical conversations. To address this, future developments could include translation memory systems or fine-tuned models capable of maintaining context across sessions. Such mechanisms would ensure that key terms, recurring phrases, and domain-specific vocabulary remain uniform throughout the interaction. These enhancements would not only elevate the overall reliability of the translation process but also make the system suitable for professional and academic use where precision is non-negotiable. Incorporating these improvements would position the system as a competitive solution in the space of intelligent, real-time communication tools. [15-18]

Conclusion

This paper introduces a comprehensive, end-to-end framework designed to enhance real-time communication through the seamless translation of spoken language across multiple tongues. The system leverages peer-to-peer video calling via WebRTC and is backed by a robust server-side pipeline that handles automatic speech recognition, language translation, and speech synthesis. Specifically, it employs Whisper for accurate speech-to-text conversion, GPT models for dynamic and context-aware language translation, and gTTS for converting translated text back into spoken audio. By uniting these technologies in a single workflow, the system addresses the ongoing challenge of facilitating effective multilingual interaction in real-time, particularly in scenarios requiring low latency and high accuracy. A key strength of the system lies in its ability to support real-time communication across five different languages. Users can either listen to translated audio or read synchronized captions directly within the video interface. The lightweight nature of the web application, combined with the scalability of its modular design, ensures that it can be deployed across a variety of platforms without demanding excessive computational resources. Furthermore, the architecture allows for the seamless integration of additional languages or components, enabling adaptability for future use cases or technological upgrades. This flexibility positions the system as a viable solution for cross-cultural dialogue in both informal and professional settings. The experimental results reveal the strong performance of the translation pipeline in multiple aspects. The system achieved high accuracy in transcribing spoken language, maintained the semantic integrity of translations, and delivered responses quickly enough to support fluid conversation. Metrics collected during testing demonstrated an average end-to-end latency of approximately 2.2 seconds, which is within acceptable bounds for live communication systems. Additionally, the caption alignment accuracy exceeded 90%, indicating that the visual transcript was well-matched to the spoken content. These findings validate the system's utility for real-world scenarios, particularly in domains that require timely and accurate multilingual interaction. Applications of the system are far-reaching and include international collaboration, remote education, inclusive conferencing, and assistive technologies for individuals with hearing impairments or language barriers. Its ability to provide both auditory and visual translation in real time enhances accessibility and ensures a smoother user experience for diverse participants. The modular and adaptable design not only supports current needs but also offers a strong foundation for future improvements, such as speaker recognition, offline processing, or support for additional languages and dialects. As global communication becomes increasingly common, this system stands as a promising step toward breaking down language barriers in digital conversations.

References

[1]. L. Alonso-Balupe and P. Romero-Fresco, "Interlingual live subtitling: the crossroads between translation, interpreting and accessibility," Universal Access in the Information Society, vol. 23, 2024.

[2]. S. Polepaka, V. P. Kumar, S. UmeshChandra, and G. Thakur, "Automated Caption Generation for Video Call with Language Translation," in Proc. of Intl. Conf. in Emerging Trends; 2023.

[3]. S. Mehta, "ML for Real-time Multilingual Communication Systems," SSRN Electronic Journal, 2018.

[4]. N. Jain, V. Kathuria, M., Sharma, V. Malik, "Real-time speech-to-text translation using machine learning, textitIEEE Access, vol. 8, pp. 145–156, 2022.

[5]. J. Liu, C. Liu, B. Shan, and O. S. Ganiyusufoglu, "A Computer-Assisted Interpreting System for Multilingual Conferences Based on Automatic Speech Recognition," IEEE Access, vol. 12, 2024.

[6]. B. Pandipati and R. P. Sam, "Speech to Text Conversion using Deep Learning Neural Net Methods," International Journal of Engineering and Applied Sciences, 2021.

[7]. S. H. Limbu, "Direct Speech to Speech Translation Using Machine Learning," M.S. thesis, Uppsala University, Dept. of IT, 2020.

[8]. V. A. Sangolgi et al., "Enhancing cross-linguistic image caption generation with Indian multilingual voice interfaces using deep learning techniques," Procedia Computer Science, vol. 226, pp. 1090–1097, 2024.

[9]. S. Amirian, K. Rasheed, T. R. Taha, and H. R. Automatic image and video caption generation with deep learning: A concise review and algorithmic overlap. IEEE Access, vol. 8, pp. 145727–145749, 2020.

[10]. N. Annamareddy et al., "Advancing Multilingual Communication: Real-Time Language Translation in Social Media

Platforms Leveraging Advanced Machine Learning Models," Journal of Chemical Health Risks, vol. 14, no. 2, 2024.

[11]. J. Singh, "Natural language processing in the real world: Text processing, analytics, and classification," Tech Chronicles, 2023.

[12]. Y. Zhang, L. Chen, and Y. Wang, "Latency-Aware Neural Translation for RealTime Bilingual Conversations," in Proc. ACL, 2022.

[13]. R. L. Wilcox and T. Schuster, "WebRTC in Real-Time Education: System Design and Evaluation," IEEE Transactions on Learning Technologies, vol. 15, no. 3, pp. 248–258, 2023.

[14]. Stan, M. Ene, and D. Popescu, "Evaluation of Text-to-Speech Engines for Multilingual Assistive Applications," Journal of Accessibility and Design for All, vol. 11, no. 1, pp. 1–21, 2021.

[15]. M. Agarwal and R. Pandey, "Real-Time Subtitle Generation Using Whisper and WebSockets," ArXiv preprint arXiv:2302.11991, 2023

[16]. M. A. Hasan and D. R. Kunze, "Scalable Audio Translation with Deep Streaming Models,," in textitProc. of INTERSPEECH, 2023.

[17]. C. Li, B. Xu, and Z. Wang, "Improving Low-Resource Language Translation using Transfer Learning and Contextual Embeddings," Neural Processing Letters, vol. 54, pp. 3095–3114, 2022.

[18]. K. Yao, T. Lei, and D. Zhang, "Real-Time AI-Based Captioning System for Web Video Conferences," in Proc. IEEE SmartTech 2022.