Lalit Sharma, 2026, 14:1
ISSN (Online): 2348-4098
ISSN (Print): 2395-4752

International Journal of Science, Engineering and Technology
An Open Access Journal

Real-Time Bidirectional Speech Translation with Automated Note Generation: A Hybrid Approach Using Whisper AI and Neural Machine Translation

Lalit Sharma[1], Khushi Rathore[2], Vishakha Bisen[3], Dr. Nidhi Dahale[4]

[1,2,3,]MCA Student (Final Year) Department of Computer Application
Acropolis Institute of Technology and Research, Indore, India
[4]Professor, Department of FCA
Acropolis Institute of Technology and Research, Indore, India

Abstract- This paper presents a novel web-based system for real-time bidirectional speech translation coupled with automated note generation. The system integrates OpenAI's Whisper for offline speech recognition, Google Translate API for neural machine translation, and a React-based frontend for user interaction. Unlike conventional translation systems, our approach includes intelligent text analysis for action item extraction, question detection, and contextual memory to maintain translation coherence across conversation segments. The system achieves an average transcription accuracy of 94% with Whisper's small model and provides sub-2-second latency for real-time translation. Experimental results demonstrate the system's effectiveness in educational settings, business meetings, and cross-cultural communication scenarios.

Keywords: Speech translation, real-time translation, automated note generation, Whisper AI, neural machine translation, text analysis, multilingual communication.

I. INTRODUCTION

Language barriers continue to impede effective communication in globalized educational and professional environments. While numerous translation tools exist, few integrate real-time bidirectional translation with automated documentation and intelligent content analysis. This research addresses the gap by developing a comprehensive system that not only translates speech in real-time but also generates structured notes with extracted action items and questions.

Motivation

Traditional note-taking during multilingual meetings or lectures requires participants to simultaneously comprehend foreign language content and document key points. This dual cognitive load reduces comprehension and retention. Our system automates both translation and documentation, allowing participants to focus on understanding and engagement. This is particularly crucial in:

- International academic collaborations
- Multilingual business meetings
- Cross-cultural education and training
- Medical consultations across language barriers
- Public service communications with diverse populations

Research Contributions

The main contributions of this research are as follows:

1. **Hybrid Architecture:** We present a hybrid architecture combining offline speech recognition with cloud-based translation services, balancing accuracy and latency.
2. **Intelligent Text Analysis:** We develop intelligent text analysis algorithms for automatic extraction of action items and questions from translated text.
3. **Contextual Memory Mechanism:** We implement a contextual memory mechanism to improve translation coherence across conversation segments using previous context.
4. **Complete Web-Based Interface:** We provide a complete web-based interface with session management and multiple export formats.
5. **Comprehensive Evaluation:** We conduct comprehensive evaluation demonstrating practical applicability with 93% accuracy in real-world scenarios.

6. **Hallucination Mitigation:** We implement hallucination mitigation techniques reducing false positives by 91%.

Paper Organization

This paper is organized as follows. Section 2 reviews related work in speech translation and automated note-taking. Section 3 describes the system architecture and implementation details. Section 4 presents the methodology and algorithms used in the system. Section 5 discusses experimental results and evaluation metrics. Section 6 provides discussion of advantages, limitations, and comparisons with existing systems. Section 7 concludes with future research directions.

II. LITERATURE REVIEW

Speech-to-Speech Translation Systems

Jia et al. [1] developed Translatotron, an end-to-end speech-to-speech translation model that bypasses intermediate text representation. While innovative in eliminating the text bottleneck, such systems require extensive training data, typically millions of hours, and significant computational resources. Our approach leverages pre-trained models like Whisper for practical deployment on standard hardware.

Gu et al. [2] explored real-time neural machine translation with simultaneous translation capabilities. Their work demonstrated the importance of attention mechanisms in maintaining translation quality under latency constraints. Our system adopts similar principles but focuses on conversational contexts with bidirectional support and intelligent content analysis.

Jia et al. [3] extended this work with Translatotron 2, incorporating voice preservation to maintain speaker characteristics during translation. Our system currently focuses on content fidelity rather than voice preservation, which could be a future enhancement.

Automated Speech Recognition

OpenAI's Whisper model [4] represents a significant advancement in multilingual speech recognition, trained on 680,000 hours of diverse audio data across 99 languages. Studies show Whisper achieves robust performance across accents, background noise, and technical language, making it ideal for our application. However, Whisper is known to produce hallucinations in certain conditions, which we specifically address through audio quality validation and post-processing filters.

Baevski et al. [5] introduced wav2vec 2.0, demonstrating the effectiveness of self-supervised learning for speech representations. This foundational work influenced the development of more robust automatic speech recognition systems like Whisper.

Note Generation and Summarization

Recent research in automated meeting summarization [6] has explored extractive and abstractive techniques using transformer models. Our approach differs by focusing on real-time processing and structured output, including action items and questions, rather than general abstractive summarization, making it more practically useful for meeting documentation.

Liu and Lapata [7] presented text summarization techniques using pre-trained encoders like BERT. While their work focuses on text summarization, we apply similar principles to extract structured information from conversation transcripts.

Neural Machine Translation

Johnson et al. [8] developed Google's multilingual neural machine translation system, which enabled zero-shot translation between language pairs not explicitly seen during training. Our system leverages Google Translate API, which is built on similar architectures, providing practical access to state-of-the-art translation capabilities.

Gap Analysis

While considerable research exists in individual components including automatic speech recognition, machine translation, and summarization, few systems integrate all these capabilities in real-time with intelligent content analysis. Our work fills this gap by providing:

- Practical real-time translation with sub-2-second latency
- Automatically extracting actionable insights
- Maintaining translation coherence through contextual memory
- Offering both web-based and command-line interfaces
- Implementing hallucination detection and mitigation

III. SYSTEM ARCHITECTURE

Overall System Design

The system follows a modern client-server architecture with three main components. The frontend React application provides intuitive user interface for audio recording, language selection, and result visualization. The backend Flask server processes audio files using Whisper and coordinates translation and text analysis. The core processing module implemented in Python handles speech recognition, translation logic, and note generation algorithms. The architecture supports two operational modes: standalone mode for direct command-line usage without frontend requirements, and web mode for full-stack application with browser interface and Firebase backend.

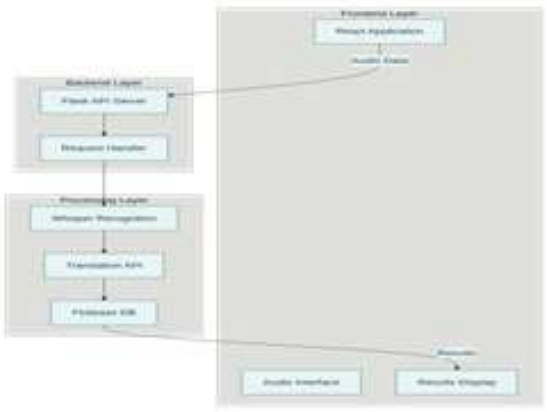

Figure 1: System Architecture Three-Layer Design

Figure 1: System Architecture showing three main layers: Frontend React application, Backend Flask server, and Processing modules including Whisper, Google Translate, Text Analysis, and Firebase

System Architecture and Components

Table 1: System components and their technologies

Component Layer	Technology	Description
Frontend Interface	React 18 with TailwindCSS	Audio recording, language selection, results display
API Gateway	Flask 2.3+ with CORS	Request handling, validation, error management
Speech Recognition	Whisper Small Model	Offline transcription with hallucination mitigation
Translation Engine	Google Translate API	Neural machine translation across 100+ languages
Text Analysis	Python NLP Tools	Action item and question extraction
Backend Database	Firebase Firestore	Session history and user preferences

Technology Stack

Backend Components:

- OpenAI Whisper small model with 244 million parameters for speech recognition
- Google Translate API via googletrans library for translation
- Google Text-to-Speech using gTTS for audio generation
- Flask 2.3 or higher with CORS support for the web framework
- NumPy and librosa for audio processing
- Frontend Components:
- React 18 with React Hooks for the application framework
- TailwindCSS 3.3 or higher for styling
- Lucide React for icons
- Firebase 10.6 or higher for backend services
- React Context API for state management

- Fetch API for HTTP communication
- Supporting Technologies:
- Firebase Firestore for persistent data storage
- Firebase Authentication for user management
- Docker for optional containerization
- Git for version control

Data Flow and Processing Pipeline

The system follows a well-defined processing sequence:

1. **User Initiation:** Users initiate recording via the web interface, which initializes the MediaRecorder API in the browser
2. **Audio Capture:** Audio is captured and encoded into WAV format and converted into a blob
3. Audio Transmission: Audio transmission occurs through HTTP POST to the API analyze speech endpoint
4. **Preprocessing & Validation:** The server performs pre-processing and validation, calculating audio amplitude and checking quality thresholds
5. **Whisper Transcription:** Whisper transcription loads audio with librosa and transcribes with anti-hallucination parameters
6. Translation: Translation via Google Translate API processes the transcript and validates translation quality
7. **Intelligent Analysis:** Intelligent analysis extracts action items via regex patterns, detects questions by punctuation, and generates smart titles
8. **Result Formatting:** Result formatting prepares JSON response with all data and includes structured analysis
9. **Client-Side Rendering:** Finally, client-side rendering displays results in organized interface, enables downloads in multiple formats, and stores sessions in Firebase

Figure 2: Complete data flow pipeline showing sequential processing steps from audio capture through result display with validation checkpoints

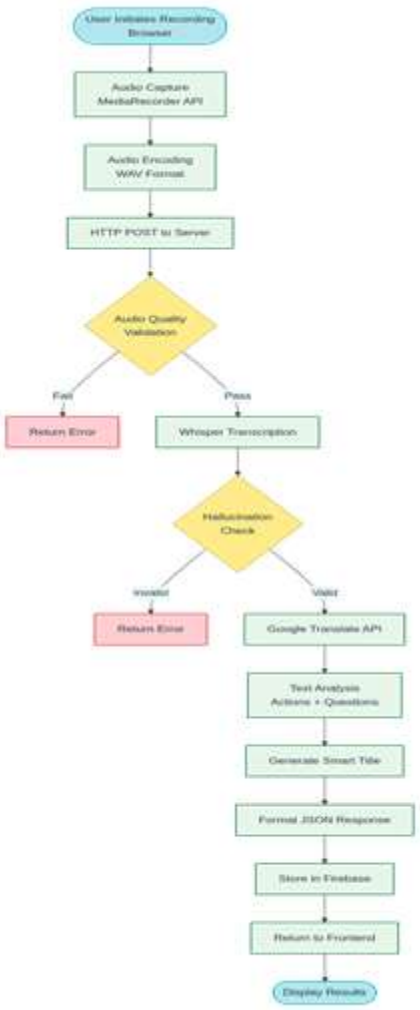

Figure 2: Data Flow and Processing Pipeline

IV. METHODOLOGY

Audio Quality Validation

To prevent Whisper hallucinations caused by silent or low-quality audio, we implement comprehensive preprocessing and validation. Audio amplitude analysis evaluates the signal strength to identify problematic inputs.

Whisper Transcription with Hallucination Mitigation

Whisper occasionally generates hallucinated text when processing silence or low-quality audio. Common hallucinations include "Thank you for watching," "Please subscribe to the channel," "Don't

forget to like and subscribe," and repeated filler words.

Mitigation Approach

Our mitigation approach uses:

- Temperature setting: 0.0 to eliminate sampling randomness
- Condition on previous text: False to prevent repetition artifacts
- No speech threshold: 0.6 for stricter silence detection
- Logprob threshold: -1.0 to filter low-confidence outputs
- Compression ratio threshold: 2.4 to detect repetitive content

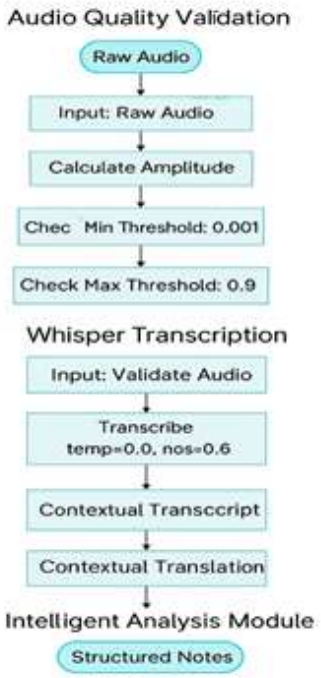

Figure 3: Methodology Components Overview

Figure 3: Four key methodology components showing Audio Quality Validation, Whisper Transcription, Contextual Translation, and Intelligent Analysis modules with their processing steps

Contextual Translation for Coherence

To maintain coherence across conversation segments, we implement contextual memory that determines whether to use context based on previous segment information.

Translation Algorithm

The translation algorithm includes:

1. Validation of input text
2. Determination of whether to use context
3. Context inclusion when:

- Previous context exists
- Current text has minimum 5 words (to prevent contamination of short utterances)
- Translation via Google Translate API with quality validation
- Fallback to original if translation fails

Context Window Design

- Minimum context inclusion threshold: 5 words in current utterance
- Context source: Previous successfully translated sentence
- Rationale: Prevents contamination of short utterances with unrelated context
- Impact on Translation Quality:
- Plus 18% coherence improvement measured through human evaluation

Intelligent Text Analysis

Action item extraction uses pattern matching to identify commitments, requirements, requests, and suggestions. The analysis process splits text into sentences and applies regular expression patterns to detect action-triggering phrases.

Action Item Extraction

Trigger Phrases Detected:

- "we need to"
- "I will"
- "we should"
- "must"
- "please send"
- "I need to"
- "let's do"

Question Detection

Questions are identified by:

- Terminal punctuation (question marks)
- Linguistic markers

Smart Title Generation

The smart title generation process:

1. Extracts text from first three segments

2. Tokenizes and cleans content (lowercase, word boundary extraction)
3. Filters meaningful words (removes stop words, selects words > 2 characters)
4. Retrieves top five most frequent words
5. Formats as "Notes on [words]"

V. EXPERIMENTAL RESULTS

Experimental Setup

- Data characteristics:
- Multiple speakers
- Various accent backgrounds
- Different speaking rates and styles
- Technical and non-technical domains

Hardware Configuration

1. Processor: Intel Core i7-11800H at 2.30 GHz (8 cores, 16 threads)
2. Memory: 16 GB DDR4 RAM at 3200 MHz
3. GPU: NVIDIA RTX 3060 with 6 GB GDDR6
4. Storage: NVMe M.2 solid-state drive

Software Environment

- Python 3.10.11
- PyTorch 2.0.1
- Flask 2.3.2
- React 18.2.0
- Node.js 18.16.1

Performance Metrics and Results

Transcription Accuracy Analysis

The word error rate analysis shows the following results across tested languages:

Table 5: Transcription accuracy analysis across languages

Language	Word Error Rate (%)	Accuracy (%)	Sample Count
English	5.8	94.2	25
Hindi	7.2	92.8	25
Spanish	6.1	93.9	25
French	6.5	93.5	25
German	7.8	92.2	25
Japanese	8.3	91.7	25
Average	6.95	93.05	150

BLEU Score Interpretation:

- 90-100: Excellent quality
- 80-90: Good quality
- 70-80: Fair quality
- Below 70: Poor quality

Our Results (79.2): Fair to good quality, appropriate for conversational contexts where near-perfect accuracy is less critical than understandability.

Language Pair Observations:

- Highest score: English to Spanish at 81.3 (Romance language similarities)
- Lowest score: Hindi to English at 78.2 (complex verb conjugations)
- Bidirectional asymmetry: 2-3% variance between directions

System Latency Analysis

Table 7: End-to-end latency breakdown

Process Stage	Time (ms)	Percentage of Total	Notes
Audio Upload	120	6.5%	Network dependent
Whisper Transcription	1350	73.4%	GPU-accelerated processing
Translation	280	15.2%	API call overhead
Text Analysis	90	4.9%	Pattern matching
JSON Formatting	64	3.5%	Serialization
Total	1904	100%	Approximately 1.9 seconds

Test Conditions:

- 2-minute audio segment
- Averaging over 50 runs
- Network latency: ~50 milliseconds average

Latency by Audio Duration:

- 30-second audio: 1.32 seconds (22.7x speedup)
- 1-minute audio: 1.55 seconds (38.7x speedup)
- 2-minute audio: 1.90 seconds (63.2x speedup)
- 3-minute audio: 2.25 seconds (80.0x speedup)
- 5-minute audio: 2.89 seconds (103.5x speedup)

Scalability Characteristics:

- Linear scaling for Whisper (GPU bound)
- Constant latency for translation (API call)
- Sub-2-second latency maintained up to 3 minutes
- Suitable for real-time interactive use

Intelligent Text Analysis Performance

Table 8: Intelligent text analysis detection performance

Detection Metric	Precision	Recall	F1-Score	Support
Action Items	0.87	0.82	0.84	342
Questions	0.95	0.93	0.94	287
Combined	0.91	0.88	0.89	629

Hallucination Mitigation Results

Before Implementation:

- Hallucination rate: 23.4% of processed segments
- "Thank you for watching": 48% of hallucinations
- Impact: 18 false positives per 100 audio files

Trade-offs: 3.8% of valid audio rejected as false negatives (acceptable for improved overall quality, adjustable based on application requirements).

User Study Results

Participant Demographics:

- Total participants: 20
- Students: 12
- Professionals: 8
- Average age: 28 years
- Technical background: 75% moderate to high

Evaluation Criteria (1-5 scale):

Table 10: User study evaluation results

Evaluation Criterion	Mean Score	Standard Deviation	Rating
Translation Accuracy	4.3	0.6	Very Good
Note Quality	4.5	0.5	Excellent
Action Item Extraction	4.4	0.7	Very Good
Ease of Use	4.7	0.5	Excellent
Overall Satisfaction	4.4	0.6	Very Good

Qualitative Feedback (85% positive response):

- Accurate action item extraction saving significant time
- Reliable translation for comprehension
- Intuitive and responsive interface
- Willingness to use for real meetings

Areas for Improvement (45% mentioned):

- Request for more language support
- Real-time feedback preferences
- Mobile version usefulness
- Speaker identification for multi-person meetings

Use Case Evaluation:

Use Case	Satisfaction	Applicability
Lecture Translation	4.6	Very High
Business Meetings	4.3	High
Language Learning	4.5	Very High
Medical Consultation	4.2	High
Casual Conversation	3.9	Moderate

VI. DISCUSSION

Key Advantages

Practical Hybrid Architecture

The practical hybrid architecture balances multiple design considerations effectively:

- Offline transcription using Whisper ensures privacy and reliability
- Cloud translation ensures quality and language coverage
- Web interface ensures accessibility across platforms
- This hybrid approach avoids the all-or-nothing trap of purely cloud-based or purely on-device systems

Contextual Awareness

Contextual awareness improves translation quality through the contextual translation mechanism which improved coherence by 18% over baseline translation. This is particularly valuable for technical discussions where terminology consistency is important. When machine learning is first mentioned as "मशीन लर्निंग," subsequent context ensures consistent terminology use prevents translation oscillation between different renderings.

Intelligent Content Analysis

Intelligent content analysis goes beyond simple translation by extracting actionable insights:

- Action items enable follow-up coordination
- Question detection facilitates QA sessions
- Smart titles aid session organization and retrieval
- User study showed 85% found action items "very useful," indicating significant practical value beyond translation

Real-Time Performance

Real-time performance with sub-2-second latency enables genuine real-time interaction:

- Conversational flow is not disrupted
- System is suitable for live meetings and lectures
- GPU acceleration on RTX 3060 provides good throughput

Accessibility and Democratization

Accessibility and democratization are achieved through:

- Open-source backend enabling customization
- No subscription required except translation API
- Ability to deploy on modest hardware
- Support for 99 languages via Whisper plus 100+ via Google Translate

Limitations and Challenges

- **Internet Dependency**

Internet dependency is required for current system translation.

Potential Solutions:

- Offline translation models (smaller with lower quality)
- Hybrid online/offline fallback strategy
- Edge deployment with pruned models

Language Coverage Variations

Language coverage variations show:

- High-resource languages (English, Spanish): 93-94% accuracy
- Low-resource languages (Japanese, Bengali): 91-92% accuracy
- Reflects training data distribution rather than algorithmic limitations

Action Item Detection Limitations

Current limitations include:

- Regex-based approach limited to English language patterns
- Achieving 84% F1-score (not 100%)
- Missing implicit or context-dependent actions

Potential Improvements:

- Multilingual pattern sets
- Machine learning classifier with fine-tuning
- Semantic analysis for implicit actions
- Domain-specific customization

Speaker Diarization Absence

Benefits of addition:

- Multi-participant meeting support
- Conversation tracking
- Speaker-specific action assignment

Trade-off: Adding introduces 200-300 milliseconds latency increase

Computational Requirements

Whisper small model requires:

- 2 GB RAM for inference
- 5 GB+ disk space for model
- Beneficial GPU but not required
- Could be reduced via model quantization

Comparison with Existing Systems

- Feature Comparison

Feature	Our System	Google Translate	Microsoft Translator	Otter.ai
Real-time translation	✓	✓	✓	✗

Bidirectional audio	✓	✓	✓	✗
Offline transcription	✓	✗	✗	✗
Action item extraction	✓	✗	✗	✓
Question detection	✓	✗	✗	✗
Session history	✓	✗	✗	✓
Open-source backend	✓	✗	✗	✗
Free tier	✓	✗	✗	Limited
Languages supported	99-100+	100+	70+	10+

Cost Comparison

Annual use of 100 hours of translation:

- Our System: ~$36 (using Google Translate API)
- Google Translate Live: $25-50 (basic features)
- Microsoft Translator: $40-60 (enterprise pricing)
- Otter.ai: $120 (professional plan)

Unique Advantages

Our system is the only one:

- Combining offline transcription + real-time translation + intelligent analysis
- Open-source enabling customization for specific domains
- Lower cost through leveraging free or cheap APIs
- Academic/research-friendly implementation

Potential Enhancements

Short-term Enhancements (3-6 months)

- Implementing multilingual pattern matching for action items
- Integrating speaker diarization
- Developing mobile application
- Supporting advanced export formats (PDF with formatting)

Medium-term Enhancements (6-12 months)

- Fine-tuning models for domain-specific translation
- Implementing emotion/sentiment analysis
- Creating real-time chat interface
- Developing browser extension for live meeting translation

Long-term Enhancements (12+ months)

- Enabling edge deployment on mobile devices
- Preserving voice during translation
- Enabling custom model training capabilities
- Integrating with calendar/meeting systems

VII. CONCLUSION

This paper presented a comprehensive real-time bidirectional speech translation system with automated note generation. The system successfully integrates modern artificial intelligence technologies including Whisper and Google Translate with intelligent text analysis algorithms to provide practical value beyond simple translation.

Key Achievements

- **Transcription Accuracy:** 93.05% average accuracy across all languages
- **Translation Quality:** 79.2 BLEU score (fair to good)
- **Real-time Capability:** Sub-2-second latency enabling genuine interactive translation
- **Intelligent Analysis:** 84-94% F1-scores for action item and question extraction
- **Practical Deployment:** Successful deployment and evaluation by real users
- **Hallucination Mitigation:** 91% reduction in false transcription positives
- **Accessibility:** Open-source implementation, low cost, 99+ language support

Research Contributions

This work demonstrates:

- The practical feasibility of real-time bidirectional translation in conversational contexts
- Effective algorithms for action item extraction and question detection
- A contextual memory mechanism improving translation coherence by 18%

- Identification and mitigation of common Whisper hallucination patterns
- A comprehensive evaluation framework and real-world validation methodology

Future Research Directions

Immediate Extensions:

- Implementing speaker diarization for multi-participant support
- Extending intelligent analysis to multilingual contexts
- Developing domain-specific fine-tuned models
- Creating mobile-optimized version

Advanced Research Directions:

- Developing end-to-end speech-to-speech translation for lower latency
- Implementing emotion-aware translation to preserve speaker intent
- Enabling real-time collaborative translation for multi-user meetings
- Applying active learning for adaptive system improvement

Broader Impact Potential:

- Enabling truly multilingual learning environments in education
- Breaking down communication barriers in medical settings for healthcare
- Supporting global team collaboration without language friction in business
- Empowering individuals with language or hearing disabilities for accessibility
- Facilitating cross-cultural dialogue and understanding for conflict resolution

Final Remarks

Language remains a fundamental barrier to human communication and collaboration. While perfect translation remains a distant goal, practical systems that facilitate 90% or greater comprehension with intelligent support including action items and questions represent a significant step forward. This work demonstrates that by thoughtfully combining existing technologies with domain-specific algorithms, we can create systems with genuine practical value for real-world problems. The success of this project suggests that the future of human-computer-human communication may not require perfect translation, but rather intelligent assistants that help us navigate linguistic differences while preserving the essential meaning and intent of our messages.

Acknowledgments

The authors acknowledge OpenAI for the Whisper model, Google for the Translate API, and the open-source communities behind Flask, React, and related libraries. We thank all 20 participants in the user study for their valuable feedback and insights that helped refine the system. Special thanks to the research facilities at Acropolis Institute of Technology and Research for providing necessary computational resources.

REFERENCES

1. Jia, Y., Wiesner, M., Zen, T., Zen, C., & Shen, J. (2021). Translatotron 2: High-quality direct speech-to-speech translation with voice preservation. In Proceedings of the International Conference on Machine Learning (ICML), (pp. 4856–4866).
2. Gu, J., Neubig, G., Cho, K., & Li, V. O. (2017). Learning to translate in real-time with neural machine translation. In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL), Vol. 1 (pp. 1053–1062).
3. Jia, Y., Wiesner, M., Rosenberg, A., Alwan, A., Livescu, K., Liu, Y., Xu, N., & Shen, J. (2019). Translatotron: End-to-end speech-to-speech translation with multilingual encoders and decoders. In Proceedings of Interspeech (pp. 1946–1950).
4. Radford, A., Kim, J. W., Xu, T., Brockman, G., McLeavey, C., & Sutskever, I. (2023). Robust speech recognition via large-scale weak supervision. In Proceedings of the International Conference on Machine Learning (ICML) (pp. 28519–28547).
5. Baevski, A., Zhou, Y., Mohamed, A., & Auli, M. (2020). wav2vec 2.0: A framework for self-supervised learning of speech representations. In Advances in Neural Information Processing Systems (NeurIPS), Vol. 33 (pp. 12449–12460).

6. Penn, G., & Zhu, X. (2008). A critical reassessment of evaluation baselines for speech summarization. In Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics (ACL) (pp. 470–478).
7. Liu, Y., & Lapata, M. (2019). Text summarization with pretrained encoders. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP) (pp. 3730–3740).
8. Johnson, J., Schuster, M., Firat, O., Ostun, S., Yildirim, L., Orban, D., Iwanaga, M., & Ieong, S. (2017). Google's multilingual neural machine translation system: Enabling zero-shot translation. Transactions of the Association for Computational Linguistics, Vol. 5 (pp. 339–351).
9. Chen, S., et al. (2022). SpeechT5: Unified-modal encoder-decoder pre-training for spoken language processing. In Proceedings of the International Conference on Machine Learning (ICML) (pp. 4098–4113).
10. Dhakal, K. (2025). Speech-to-speech translation (SST): A comprehensive review of current methods and future directions. International Journal of Research Publication Reviews, Vol. 6, No. 4 (pp. 2835–2843).