Real Time Speech-to-Speech Translation System

Rueben Varghese Philip
School of Computer Science Engineering (SCOPE)
Vellore Institute of Technology
Chennai, India
ruebenvarghese.philip2021@vitstudent.ac.in

Vineetha C
School of Computer Science Engineering (SCOPE)
Vellore Institute of Technology
Chennai, India
vineetha.c2021@vitstudent.ac.in

Srihari Gopal Karthikayini
School of Computer Science Engineering (SCOPE)
Vellore Institute of Technology
Chennai, India
sriharigopal.karthi2021@vitstudent.ac.in

***Abstract* — The ability to communicate effectively between speakers who have different languages faces obstacles due to linguistic differences. The achievement of real-time speech-to-speech translation remains complex because of multiple speech variations including pronunciation, accents, dialects, and contextual understanding though translation tools at present mostly focus on text-based translation. The outcome success of current systems suffers from both high delays and context-diminished translation capabilities. The main purpose of this project is to make an immediate speech-to-speech translator that unites three fundamental components: automatic speech recognition (ASR), machine translation (MT), and text-to-speech synthesis (TTS). Through its system the solution provides good language translations with minimal delay for allowing natural flow of dialogs across language barriers. The solution provides enhanced interaction during conversations, customer service and medical consultations.**

I. Introduction

Companies and others require effective cross-lingual communication in the present globalized world while language barriers stop organizations from collaborating efficiently. The promising solution against language barriers in communication is speech-to-speech translation (S2ST). The process of speech-to-speech translation (S2ST) fulfils three requirements which are automatic speech recognition (ASR), machine translation (MT) and text-to-speech (TTS). Real- time communication needs systems with high accuracy capabilities combined with high speed and fluent results. The efficient operation of S2ST requires neural machine translation models with excellent quality as well as minimized latency and optimized inference pipelines.

The process of speech-to-speech translation faces many difficulties related to varying speech patterns and poor recording quality as well as semantic and contextual analysis requirements, performance speed limitations and language support limitations, speech output quality demands. The natural speech contains accent variations together with various pronunciation options followed by interruption but real-time translation needs quick and precise inference at low latency for accurate results. Real-time translation models need additional processing time while context together with semantics prove essential for their operation. The technical aspects of developing fluent real-time natural speech prove difficult particularly when working with languages that lack abundant resources available.

This project implements three sequential real-time S2ST stages which address speech recognition difficulties. A transformer-based model inside the ASR component transforms spoken audio into text output. The machine Translation (MT) component runs a custom transformer architecture which received training through Opus datasets for Tamil, Hindi and English. The TTS segment transforms translated words into speech through neural technology which produces natural voice along with appropriate prosodic qualities.

II. Literature Review

There are various research papers which explores the study of Speech-to-Text (STT) and Machine Translation (MT) integration for Arabic/English broadcast news translation from spoken content. The approach aims to solve multiple issues in speech-to-text work including segmentation of sentences and the restoration of punctuation together with STT performance and transformation of spoken numbers into text format.

The STT/MT system executes data through a sequential pipeline mechanism that depends on 1-best STT output and determines its performance via Translation Edit Rate (TER). The performance of MT systems shows stronger potential to improve Translation Edit Rate than the accuracy level of STT. The combination of automatic audio parameters with HELM models for SBD segmentation produces superior translation results. The introduction of spoken number conversion technology leads to a minimal decrease in Translation Edit Rate. The adjustment of MT decoding weights on broadcast news information leads to higher accuracy levels.

The paper investigates end-to-end speech-to-text translation from English to Russian by using deep recurrent neural networks (RNNs) through Long Short-Term Memory (LSTM) models. The system requires three stages for modeling before executing a search while learning takes place in the middle. Adam optimizer delivered superior results although the text-to-text approaches showed effectiveness except when facing challenges from restricted datasets.

A portable system renders speech-to-speech translation through a PDA platform on embedded Linux to provide almost immediate spoken English to Mandarin Chinese conversions. Protective Industries established the ATR multilingual S2ST system that performs real-time translation between English-Japanese-Chinese through a combination of speech recognition mechanics together with machine translation and text-to-speech synthesis modules.

The research establishes voice-to-voice machine translation as an operational solution for clinical environments to tackle interpretational communication challenges. The Real-Time Voice Translator (RTVT) integrates Automatic Speech Recognition (ASR) with Machine Translation (MT) and Text-to-Speech (TTS) under Natural Language Processing (NLP) to form a single operation.

The SPEECHTRANS system which Carnegie Mellon University developed functions as a real-time mechanism to translate speech into speech primarily for conditions with noisy and speaker-independent speech inputs. The main functional elements of this system are phoneme extraction followed by error-correcting parsing and a confusion matrix used to select the best hypothesis.

The PolyVoice language model system delivers S2ST functionality through an approach which enhances translation quality and speaker-preserving listening quality and natural audio outcomes. The technology transforms both spoken and unspoken languages by handling discretized speech units which emerge from automatic knowledge acquisition. VALL-E X serves as the speech synthesis engine to retain speaker characteristics and avoids storing texts during the process. The system delivers exceptional speech clarity and demonstrates exceptional results in various speech processing operations such as ASR, ST, MT, and TTS.

When users deposit their voice into the system it converts speech to text and voice into the target language through expressive output with incorporated emotional elements. The application runs on Python and Spyder platform together with Google Translate API while containing four functional modules: Login, Input, Translation and Output. Through a graphical user interface administrators and end-users can access the system which handles text conversion by recognizing speech input.

This NMT framework implements reinforcement learning to perform simultaneous translation while evaluating both BLEU score and translation delay through its reward function. The system displays superior performance during experiments involving English-Russian and English-German pairs since it produces better results at lesser delay than segmentation-based techniques and NMT benchmarks.

Speech to-Speech Translation (S2ST) technology encounters problems stemmed from varied languages alongside restricted dataset availability together with restrictions imposed by the system. Research directions for the future should focus on testing S2ST systems through large field and social programs as well as developing enhanced conversational models for disfluent speech, dynamic language model creation, platform interface standardization and multi-language expansions, and lastly resolving privacy and copyrights issues.

III. Proposed Methodology

To construct real-time speech-to-speech translation the framework, it requires three interconnected models which are Automatic Speech Recognition (ASR), Machine Translation

(MT), and Text-to-Speech Synthesis (TTS). The framework contains a structured process through which speech inputs get translated into English and Hindi and Tamil. For example, the system functions through ASR speech transcription which MT transforms into Tamil output before TTS produces synthetic speech. The approach uses several integrative steps to enable translation functionality while improving spoken flow and delivering live language exchanges between the three languages.

ASR functions as a fundamental element of real-time translation systems which performs speech-to-text conversion before Machine Translation components handle the processed text. Preprocessing alongside feature extraction together with modeling procedures in ASR systems result in high accuracy alongside reliability for transcribing purposes. The accomplishment of ASR as a system stems from its effective identification of spoken words. The advancement of deep learning techniques led to new ASR model development through use of neural networks which include Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) Networks and Transformer-based models. The latest version of Automatic Speech Recognition systems represented by Wav2Vec, Whisper and DeepSpeech deploy modern advancements to provide exceptional speech recognition performance. Except for fine-tuning with minimal labeled information Wav2Vec proves its ability to work with the Mozilla Common Voice dataset immediately. The Wav2Vec ASR model with fine-tuning capabilities includes four main operational parts starting with Audio Preprocessing and moving to Feature Extraction followed by Language Modeling (Fine-tuning and Training) after which comes post-processing. Training occurs with AdamW optimizer through mixed precision optimization to accelerate convergence speed.

Another essential component of real-time speech-to-speech operations is Machine Translation. The system delivers multilingual translation services which operate between English, Hindi, and Tamil for bidirectional translation to enable fluent communication. Modern MT models consist of Sequence-to-Sequence (Seq2Seq) models as well as transformer-based models and the multilingual models mBART and M2M-100. The system implements mBART-large-50 which serves as a multilingual transformer that operates using 50 languages for English-Hindi and English-Tamil translation tasks. The system functions by processing data, building its architecture, adjusting parameters through fine-tuning and training the models. It starts with the data preparation by selecting their preferred dataset and then divide text into tokens before extending sentences to an equal length when necessary. The mBART-large-50 model forms the base architecture before receiving additional fine-tuning for its ability to translate English and Hindi and English and Tamil in both directions. The training method starts by initializing the model then adjusts its parameters through gradient accumulation during which evaluation occurs at each epoch and saves checkpoints.

Text-To-Speech Synthesis (TTS) also functions as a fundamental component in real-time speech-to-speech translation systems through its conversion of plain fluent understandable text to mechanical sound. To generate quality synthetic speech TTS needs advanced linguistic processing together with acoustic modeling and waveform generation techniques. The quality of TTS effectiveness depends on achieving human-like performance in pronunciation together with suitable rhythm and expressive delivery. The traditional TTS framework comprises four main operational elements starting with text preprocessing followed by linguistic feature extraction after which acoustic modeling happens before waveform generation completes the process. The advancement of speech synthesis through modern TTS models Tacotron 2, FastSpeech and VITS is realized through attention mechanisms in addition to improved training speed and inference response as well as the implementation of diffusion-based speech synthesis. The system adopts ParlerTTS for speech synthesis because it offers both effective voice production and good speech quality features.

A ParlerTTS-based TTS model refined through the translation system divides its operations into four main sections which include data preparation and preprocessing and fine-tuning and training and inference and deployment and speech clarity enhancement. Through diffusion-based learning the model performs multiple iterations of audio refinement before training takes place inside an environment that utilizes GPU acceleration.

Real-Time Speech-to-Speech Translation System operates as a fluid pathway through which input speech moves from acquisition to speech processing and delivers real-time translated speech. The system processes natural speech through its input speech acquisition stage before it moves to ASR processing followed by text translation along with tokenization until it reaches the stage of speech synthesis. Input speech passes through the ASR module where it becomes text before the MT module receives it for processing. After receiving text input from the MT module, the translation functions start to work alongside tokenization followed by sending translated text to the TTS module. The system delivers the translated speech during real-time operation.

IV. Experimental Setup

Data Collection and Preprocessing

ASR (Automatic Speech Recognition) requires speech recordings together with their text transcriptions to function properly. The recordings originate from speech corpora that are publicly accessible and proprietary datasets. The audio input undergoes data cleanup through noise filtering and volume regulation and then generates feature data such as MFCC level (Mel- Frequency Cepstral Coefficients) and spectrogram output. Speed perturbation and pitch shifting and background noise addition as well as other data augmentation methods strengthen model robustness.

The collection of MT (Machine Translation) data involves obtaining bilingual parallel corpora which contain text information in both source language and target language. The dataset undergoes data cleaning through duplicate removal and translation incompleteness filtering while text normalization includes converting uppercase to lowercase and removing special characters. The process of tokenization requires subword techniques including Byte Pair Encoding (BPE) and SentencePiece to address out-of-vocabulary words. These are done using the MBart tokenizer. The dataset is directly accessed from Opus via Huggingface.

The dataset for TTS (Text-to-Speech) includes pairs of text and high-quality speech recordings. Text normalization (expanding abbreviations as well as numbers and symbols into readable words) and phoneme conversion and prosody annotation occur within the preprocessing stage of the process. The audio processing includes silence trimming followed by volume normalization and feature extraction methods that generate Mel spectrograms.

Model Architecture

The ASR model operates through encoder-decoder structures that implement attention mechanisms or CTC-based models. Three typical model architectures for ASR applications are Wav2Vec 2.0, DeepSpeech and Transformer-based models. Here the first model is chosen. The input waveform features are extracted by the encoder component before the decoder transforms them into textual outputs.

The MT model operates using Transformer architectures including MarianMT and OpenNMT. The models use self-attention processing together with encoder-decoder structures which process source language text through the encoder and produce translated text from the decoder. Domain-specific training processes here occur through the application of pre- trained model mBART in this project.

The TTS model implements ParlerTTS as its main architectural component. After transforming input text to phonetic or character-based symbols through a text encoder the system generates spectrograms.

Training Setup

The training process of ASR relies on GPU acceleration technology to work with extensive speech dataset information. The training pipeline employs supervised learning with CTC loss or sequence-to-sequence loss with attention as its loss functions. The training process requires multiple iterations through the data while selecting a batch size that matches GPU memory capacity.

MT training is based on fine-tuning the mBART-50 model. The training process occurs on parallel text data which consist of pairs of languages.

The model of TTS was fine-tuned on a customized version of the IndicTTS-Hindi and Tamil dataset, which was enhanced with speaker-specific and environmental metadata. The ParlerTTS model is a prompt-driven, multi-modal text-to-speech synthesis architecture. Similarly, it was done for Tamil and English as well.

Evaluation Metrics

ASR systems are evaluated through Word Error Rate (WER) and Character Error Rate (CER). WER evaluates the percentage of misidentified words but CER shows transcription accuracy at the character level. Lower values indicate better performance.

MT evaluation relies on BLEU (Bilingual Evaluation Understudy). BLEU evaluates machine translations by comparing them to references through n-gram overlap analysis.

The assessment tool for TTS systems is simple human evaluation. Based on the speech output, the model is ranked good or bad.

ASR, MT, and TTS need to be loaded early to generate their respective outputs.

V. Results and Analysis

The evaluation of the ASR model centers on understanding its ability to convert speech into text accurately and reliably. The Word Error Rate (WER) serves as the primary assessment metric to measure transcription errors' percentage. The model shows varying performance levels based on the combination of different datasets and speaker accents and it produces higher Word Error Rates when processing noisy speech or when dealing with speakers who have pronounced regional accents.

Using the saved model and processor, they are imported and using the transcribe_audio function, we can generate the transcription by providing the audio recording directly.

The model successfully transcribes the audio and generates the transcript and displays the output. The transcription success rate is mostly 90% with a minute number of errors present.

It is observed that the training loss is about 25% which means training accuracy is about 75%, the validation loss is about 45% which means validation accuracy is 55% and Word Error Rate is 30%, which means Word Accuracy is 70%.

The Wav2Vec-tuned ASR optimized model scores ~80% accuracy and can thus be applied for real-time translation of speech. Its performance depends on hardware limitations, training time, data quantity, and voice variability. The model performs with a WER of ~20%, which is encouraging but cannot handle deep accent and noisy environments. Implementing the model is challenging, especially in terms of hardware since fine-tuning requires high-end GPUs (16GB+ VRAM).

It requires a lot of time to train on large datasets such as Mozilla Common Voice, and though SSL minimizes the requirement for labeled data, high-quality transcripts are still needed. The model is also not good at non-standard accents and low-resource languages and needs more fine-tuning using diverse datasets. Optimizations like quantization and distillation can compress model size to deploy on less powerful hardware. Denoising autoencoders and spectral subtraction can enhance noisy environment performance. A multi-stage fine-tuning approach—pre-training on general datasets followed by domain-specific training—can help improve adaptability.

The machine translation processes are evaluated through a standard evaluation metric, BLEU score. The BLEU score uses n-gram comparisons to measure translation quality regarding matched terms between computer-generated text and human references while showing variations in its results based on the language difficulty level. Structurally similar language pairs tend to produce better results from the model yet distant language pairs struggle when they possess strong grammatical dissimilarities.

The training process of the mBART-large-50 model on the English to Tamil (en-ta) concluded after 5 epochs for Tamil (and 10 for Hindi), completing 7095 training steps in approximately 2 hours and 53 minutes. The recorded final training loss was 0.8055, indicating a stable convergence of the model. The training efficiency metrics reveal that the model processed approximately 10.9 samples per second and achieved a step processing speed of 0.681 steps per second. The total computational effort for the training process was approximately 3.07e+16 floating-point operations (FLOPs), showing the extensive computational requirements of transformer-based architectures.

During the evaluation phase, the model achieved a validation loss of 0.2609 at the final epoch, with a lowest recorded validation loss of 0.2507 at epoch 3. The evaluation process was completed in 57.49 seconds, where the model processed 34.79 samples per second and executed 4.35 steps per second, demonstrating an efficient inference speed. The final BLEU score obtained was 13.64, calculated based on n-gram precision matching between the predicted and reference translations. The brevity penalty (BP) of 0.9217 indicates that the model produced slightly shorter translations than the reference, contributing to a marginal reduction in the overall BLEU score.

The model learned translation patterns successfully from English-Tamil to Hindi-English, with training loss decreasing from 3.54 (epoch 1) to 0.1867 (epoch 5). Validation loss leveled off after epoch 3 (0.2507), indicating increased training has a chance to overfit. A BLEU score of 13.64 reflects moderate translation accuracy but is poor on longer n-grams (1-gram: 43.28%, 2-gram: 18.45%, 3-gram: 9.96%, 4-gram: 6.03%). The brevity penalty (0.9217) indicates shorter outputs, which may miss context and could be resolved by extending sequence length or modifying decoding with a broader beam search.

The fine-tuned ParlerTTS model was trained on a combination of both the LJSpeech English dataset and the IndicTTS Hindi and Tamil Dataset. The model successfully generates the audio from the given input text that is transcribed and translated from the ASR and MT Model respectively. An Optional Speaker description can be given to set the condition for the audio to be generated in that speaker's tone or setting thereby making the audio sound more humane although, the audio is generated with some minute errors.

The fine-tuned ParlerTTS model performs well but requires further training for more accurate and natural speech. Training loss decreased steadily to 8.79, while evaluation loss stabilized at 4.01, demonstrating effective learning despite the multilingual nature of the task. Codebook losses ranged from 2.95 to 4.21 (evaluation) and 6.92 to 9.27 (training), confirming strong latent representation learning for phonetic translation across languages. A stable gradient norm (0.21) and a learning rate of 8e-5 facilitated smooth optimization. Codebooks, essential for encoding and decoding speech features, exhibited effective quantizer convergence across all nine codebooks. The eval/clap metric of 0.239, though modest, validates semantically aligned latent audio synthesis. Overall, the model effectively captures shared and distinct acoustic features in English, Hindi, and Tamil, but improvements in intelligibility and code-switching accuracy remain areas for enhancement.

All three models share similar obstacles which primarily consist of data quality concerns along with domain adjustment and processing speed efficiency demands. The noise-resistant training techniques benefit the ASR model but the MT model requires enhanced rare word processing alongside context-oriented improvements and the TTS model would benefit from improved prosody management to produce expressive speech.

Better speech-to-speech translation applications could be achieved through future improvements which integrate the training of ASR, MT, and TTS models as a unified system. Self-supervised learning techniques combined with larger pre-trained models serve to boost performance in all related tasks.

VI. CHALLENGES IN REAL-WORLD SCENARIO

Real-time systems implementing S2ST need systems operating with low latency together with high accuracy along with robust quality combined with multilingual support and the capability to scale while delivering efficient error management solutions. The system components need to handle speech input accurately while dealing with different accents and noise and producing natural-sounding translations that also read naturally from text. The system needs extensive training on various datasets that also includes diverse content, understand the sentiment and context of the sentence spoken as well as efficient processing capabilities and automated error detection and correction systems. The implementation of real-time S2ST faces four primary challenges which consist of conflicting latency versus accuracy levels as well as garbled speech signals combined with difficulty in sustaining both context integrity and natural speaking quality and excessive resource usage. The relationship between low latency and processing time accuracy is inversely proportional yet high-accuracy models require more time to process. The entire system pipeline becomes vulnerable to degradation when noisy speech enters the system.

The maintenance of contextual clarity with resource-friendly processing and memory optimization defines real-time application requirements on either edge systems or cloud environments.

VII. CONCLUSION

The developed system operated in real time by uniting ASR with MT through a pipeline alongside TTS. The system operates with multilingual datasets which include Common Voice and Opus-100 and LJSpeech and IndicTTS to process English and Tamil and Hindi languages. Transformer models serve as the base architectures for ASR and MT systems and team up with ParlerTTS model to create both effective speech translation solutions and speech synthesis outputs. Live speech translation capabilities of the system show how communication barriers dissolve when people speak real-time. The system overcomes preprocessing dataset issues and extensive model development period to deliver acceptable translation results and speech generation. The system can scale to different real-world applications such as healthcare and education and customer support and multilingual virtual assistants. The system provides a basis for developing emotion-aware speech translation capabilities which maintains the emotional characteristics of the original speaker.

REFERENCES

[1] Matsoukas, S., Bulyko, I., Xiang, B., Nguyen, K., Schwartz, R., & Makhoul, J. (2007, April). Integrating speech recognition and machine translation. In 2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07 (Vol. 4, pp. IV-1281). IEEE.

[2] Gibadullin, R. F., Perukhin, M. Y., & Ilin, A. V. (2021, May). Speech recognition and machine translation using neural networks. In 2021 International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM) (pp. 398-403). IEEE.

[3] Zhou, B., Gao, Y., Sorensen, J., Déchelotte, D., & Picheny, M. (2003, November). A hand-held speech-to-speech translation system. In 2003 IEEE Workshop on Automatic Speech Recognition and Understanding (IEEE Cat. No. 03EX721) (pp. 664-669). IEEE.

[4] Nakamura, S., Markov, K., Nakaiwa, H., Kikui, G. I., Kawai, H., Jitsuhiro, T., ... & Yamamoto, S. (2006). The ATR multilingual speech-to-speech translation system. IEEE Transactions on Audio, Speech, and Language Processing, 14(2), 365-376.

[5] Tzoukermann, E., & Miller, C. (2018, March). Evaluating automatic speech recognition in translation. In Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (Volume 2: User Track) (pp. 294-302).

[6] Hudelson, P., & Chappuis, F. (2024). Using Voice-to-Voice Machine Translation to Overcome Language Barriers in Clinical Communication: An Exploratory Study. Journal of General Internal Medicine, 1-8.

[7] Limbu, S. H. (2020). Direct Speech to Speech Translation Using Machine Learning.

[8] D. Shilvant, E. Sayyed, S. Jagdale, Prof. R. Waghmare (2024), Real Time Voice Translator, 42, Volume 4 Issue 1, April 2024, International Journal of Advanced Research in Science, Communication and Technology. (n.d.). Naksh Solutions. https://doi.org/10.48175/568

[9] V. Kharat, U. Chaudhari, K. Kesarwani (2022), Regional Language Translator Using Neural Machine Translation, 2, Volume 9 Issue 5 2022, International Journal Of Current Engineering And Scientific Research. https://doi.org/10.21276/ijcesr

[10] Vyas, R., Joshi, K., Sutar, H., & Nagarhalli, T. P. (2020, March). Real time machine translation system for english to indian language. In 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS) (pp. 838-842). IEEE.

[11] Dhawan, S. (2022). Speech to speech translation: Challenges and future. International Journal of Computer Applications Technology and Research, 11(03), 36-55.

[12] Tomita, M., Tomabechi, H., & Saito, H. (1990). Speech Trans: An Experimental Real-Time Speech-To-Speech Translation System. 어학연구.

[13] S. Chaudhari, A. Shukla, T. Gaware (2022), Real Time Direct Speech-to-Speech Translation, 104, Volume 9 Issue 1. January 2022, International Research Journal of Engineering and Technology. E-ISSN: 2395-0056

[14] Dong, Q., Huang, Z., Tian, Q., Xu, C., Ko, T., Zhao, Y., ... & Wang, Y. (2023). Polyvoice: Language models for speech to speech translation. arXiv preprint arXiv:2306.02982.

[15] Gu, J., Neubig, G., Cho, K., & Li, V. O. K. (2017). Learning to Translate in Real-time with Neural Machine Translation. Conference of the European Chapter of the Association for Computational Linguistics, 1, 1053–1062. https://doi.org/10.18653/V1/E17-1099