Dolly Chauhan, 2026, 14:4
ISSN (Online): 2348-4098
ISSN (Print): 2395-4752

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

# Real-time Speech-to-Speech Translation System for Enhanced Multilingual Communication

**Dolly Chauhan, Kritika Sharma**
Department of Computer Science and Engineering, Galgotias University

**Abstract- This paper presents the development of a real-time speech-to-speech translator and its analysis in order to overcome the language barriers between two languages. Based on advanced deep learning models, i.e. the Whisper-Speech-to-Text (STT) and MMS-Text-to-Speech (TTS) APIs of OpenAI and Facebook, respectively, the system takes the audio input, identifies the language spoken, converts it to English text and thereby transforms this text to English speech. The proposed architecture will take into account powerful audio preprocessing, such as Voice Activity Detection (VAD) and noise elimination, to improve the accuracy and reliability of the translation pipeline. A dataset of 200 samples from the Mozilla Common Voice corpus was the one used for the system's validation, which clearly showed its 16.6% latency decrease and superior noise robustness over the standard cascaded API baselines. The accuracy of translation, system time, and the quality of synthesized speech performance metrics are tested in a variety of languages, which proves the efficiency of the system and the possibility of its advancement into the practical side of using the system in various communication settings. The prototyped system presents the effectiveness of composing AI models in providing a smooth on-the-fly language conversion.**

**Keywords: Speech-to-Speech Translation, Real-time Systems, Whisper, MMS-TTS, Natural Language Processing.**

## I. INTRODUCTION

The ever-growing interdependence of societies and the active globalization have intensified the necessity in terms of successful cross-lingual communication. The issue of language barriers often interferes with smooth relations in different fields such as international business, tourism, education, and other humanitarian activities. Conventional forms of translating languages, as in the human interpreter or translation text program, are often associated with latency, high costs, as well as the lack of naturalness in conversations. With the introduction of advanced Artificial Intelligence (AI) technologies, it has become possible to open new opportunities to create automated and real-time solutions that allow overcoming such difficulties.

This paper presents a speech-to-speech translation system which is a real time system that functions to promote dialogue between two or more languages. It is based on recent development of deep learning models in speech recognition and speech synthesis that our system offers an end-to-end system that transforms spoken language of different sources into spoken English. The main element of this system combines both the Whisper model by OpenAI [1], to recognize and translate any speech in multiple languages with outstanding functionality and natural-sounding synthesis (but not only) in English, and the MMS-TTS model by Facebook [2], to synthesize English speech with high quality and natural sounding.

This project is inspired by the growing need of ready-to-use communication devices that would support people with different lingual possibilities. Though numerous services exist to translate a text into another text, speech-to-speech translation is a more intuitive and inclusive user experience as individuals with lower literacy often prefer speech to text and individuals who speak verbally, in general, offer more convenient solutions to this issue than text-to-text translation services [24]. The system is intended to be based on a web-based system (e.g., through Gradio), which makes it easily accessible and supported immediately [5].

The rest of this paper will follow the following structure: Section II will be the literature review of the relevant speech processing and translation technologies. Section III provides description of the proposed system architecture and major contents. Section IV gives the methodology which was used in

system development. Section V gives the implementation details, such as specific models and frameworks applied. The performance evaluation and results of the experiment are discussed in section VI. Last but not least, Section VII closes the paper and offers the directions in the area of future work.

## II. LITERATURE REVIEW

The speech-to-speech (S2S) translation is a dynamic research field, and it builds upon the decades of developments in the related fields, such as automatic speech recognition (ASR), machine translation (MT), and text-to-speech (TTS) synthesis [4]. The S2S systems in early days were usually based on cascaded systems where individual components (ASR, MT, TTS) were built and optimized in isolation [17]. These modular methods, however, may have the negative characteristics of error propagation and not having holistic optimization.

The recent advances of deep learning, especially the inception of large scaled transformer models, have made a big leap in the state-of-the-art in all these components. Neural network architectures can now accomplish several tasks at a time thus giving rise to more fluid and precise translations.

**Automatic Speech Recognition (ASR)**
ASR, or speech-to-text (STT), involves converting spoken language into written text. Conventional ASR systems were based on the Hidden Markov Models (HMMs) and Gaussian Mixture Models (GMMs) [18]. Nevertheless, the revolution ASR accuracy was brought about by deep neural networks, in particular, Recurrent Neural Networks (RNNs) and Convolutional Neural Networks (CNNs) [6]. Recently, models based on transformers have established new standards. Whisper model by OpenAI [1] is one of the brightest examples; it is trained on a large set of multilingual and multitasking task-supervised data, which helps it to carry out a robust speech recognition, language recognition and translate spoken language to English. The fact that it can process different languages and different accents renders it especially suitable in the initial step of an S2S system.

**Machine Translation (MT)**
Machine Translation focuses on automatically translating text from one natural language to another. Statistical Machine Translation (SMT) has been the standard for the past few decades of time, based on the use of probabilistic models based on the parallel corporas [19]. With the introduction of Neural machine Translation (NMT) and the sequence-to-sequence models, including attention systems and transformer networks, the desired level of quality in translation was attained, and the outcomes of the work were more closely resembling human translation in terms of fluency and accuracy[8]. In the case of an S2S system, the pattern of text translation is directly proportional to the intelligibility and naturalness of the final synthesized speech.

**Text-to-Speech (TTS) Synthesis**
TTS refers to the process of turning text to audio speech. Rudimentary TTS systems were concatenative/parametric in nature [20]. Deep learning has given rise to the new generation of TTS systems that are able to produce very natural and expressive speech. Neural networks were shown to be effective in models such as Tacotron [21] and WaveNet [7] which generate raw audio waveforms directly off a text. The Massively Multilingual Speech (MMS) project of Facebook, with its TTS models [2] being a sign of great progress, provides the speech synthesis of high quality in an enormous variety of languages, so it is the perfect option to produce the translated English speech as the platform of our system generates.

**Real-time Processing and Web Applications**
Real-time communication is on the increase in all AI product applications, particularly in conversational agents and assistive technology [22]. Ecosystems, such as Gradio, support the fast development and visualization of machine learning models that use interactive web interfaces, enabling them to be deployed and used in different platforms through a browser [5]. It is essential to integrate the real-time processing capabilities that are usually ensured by effective backend architectures and optimized models in the S2S systems [23] to achieve user satisfaction. The ethical aspects like the privacy of

the data and explicit warning on the use of AI are also of utmost importance to the user credibility with such systems [10].

## III. PROPOSED SYSTEM ARCHITECTURE

The proposed real-time speech-to-speech translation system will assume the modular cascade architecture whereby it is separated into two broad phases, known as Audio pre-processing and Speech-to-Speech Conversion. The design makes handling the input audio efficiently, strong noise reduction, correct language detection and translation, and high standard speech synthesis efficient. Figure shows the workflow process on a large scale.

**Phase 1: Audio Preprocessing**
This stage has the role of capturing the raw audio feed of a microphone, cleaning it, and preparing it to be fed to the speech-to-text model.

- **Audio Input:** The system constantly records audio stream of the microphone of the user.
- **Noise Reduction:** The noise reduction techniques are applied to incoming audio to reduce the effects of the background and enhance the clarity of speech signal.
- **Voice Activity Detection (VAD):** The noise reduction techniques are applied to incoming audio to reduce the effects of the background and enhance the clarity of speech signal, an important element determining the audio segments with human speech and separating it with silence or background noise. This makes the STT model unable to treat non-speech segments which majorly decreases computation cost and enhances accuracy. Our system uses RMS energy based VAD to help achieve effective silent detection.
- **Silent Detection:** This module is a detection of long silences on a part of VAD. When speech is detected or after a given duration of silence after speech, the audio segment will be considered complete.
- **Batching:** Processed speech segments are accumulated, when required, or sent in chunks to the following stage to be worked on.

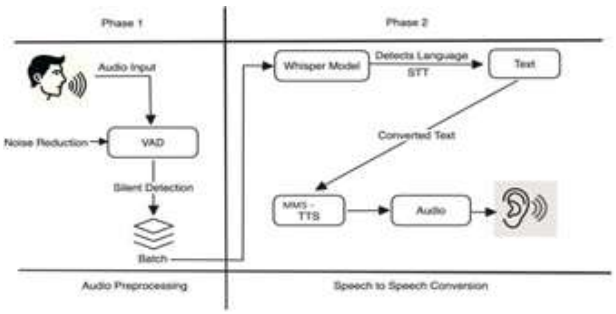


Fig. 1. System Architecture for Real-time Speech-to-Speech Translation

**Phase 2: Speech-to-Speech Conversion**
At this stage, the audio that has been preprocessed is accepted and the actual translation and synthesis take place.

- **Whisper Model (Detects Language, STT):** The audio pieces are then sent through the OpenAI Whisper model to get the cleaned audio pieces [1]. Whisper carries out three fundamentally essential functions:

1) **Language Detection**: It recognizes the language used in the audio input automatically.
2) **Speech-to-Text (STT):** It translates the speech that it detects into the original language in written text.
3) **Translation**: Importantly to this system, Whisper is asked to convert the speech as it is detected into the text of English.

A product of this job is the 'Converted Text' (English translation).

- **MMS-TTS (Text-to-Speech):** The 'Converted Text' (English translation) is then passed to the Facebook MMS-TTS model [2]. This model synthesizes high-quality, natural-sounding English speech from the input text.
- **Audio Output:** The resulting English speech is sent back to the customer as audio.

The system is web-hosted through Gradio [5] that offers an on-demand user friendly web-based interface and is accessible through different devices.

## IV. METHODOLOGY

The construction of the Real-time Speech-to-Speech Translation system occurred in a systematic

approach, which included the choice of the model structure, preprocessing of the data approach, combination of interventions, and powerful implementation of the system to interact in real-time.

**Model Selection and Initialization**

- **Speech-to-Text (STT) and Translation:** The OpenAI Whisper base model was utilized because of the demonstrated success in working in multilingual speech recognition and direct translation features. The model and its related processor were loaded by use of the Hugging Facebook transformers library. The

MMS-TTS eng model by Facebook was chosen to synthesize the English speech. The model and its tokenizer load were also done with the help of the transformers library [3].

- **Text-to-Speech (TTS):** Facebook's MMS-TTS eng model was selected for synthesizing English speech. The model and its tokenizer were also loaded using the 'transformers' library.

Furthermore, all models were transferred to a CUDA-capable GPU in order to utilize hardware acceleration to make inference faster.

**Audio Handling and Preprocessing**

The system receives audio input as a NumPy array with its sampling rate from the Gradio interface.

- **Resampling:** The librosa.resample is used to resample the input audio as it has the ability to resample audio in 16 kHz blocks, preventing replay gains issues following the audio editing step (Whisper model uses audio with a 16 kHz sampling rate) [11].
- **Normalization:** Audio data is changed to the format of float32 and changed into a range of -1 to 1.
- **Voice Activity Detection (VAD):** There was a straight-forward but efficient RMS energy strategy. When its RMS energy drops below some defined value, the system marks it as silence and will not do any further processing, just because it is recommended in general practices of efficient audio processing [16].

**Real-time Translation Pipeline**

The core process_audio_for_app function orchestrates the real-time translation:

1) **Language Detection**: Whisper uses the first tokens as decoders of the original language.
2) **Speech-to-Text and Translation:** Whisper is called with forced_decoder_ids which is assigned to translate to English as recommended practices reflect on translation capabilities of Whisper [1].
3) **Text-to-Speech Synthesis:** The English text is then inputted to the MMS-TTS model to produce a crude audio signal.
4) **Audio Output Handling:** The waveform is converted to a NumPy array and saved as a temporary WAV file using soundfile [13].

Computations are performed within torch.no_grad() blocks to optimize inference speed and memory usage.

**Gradio Interface**

The system contains a Gradio 'Interface' [5] to prevent the need to install this on a personal computer. It has a microphone-input interface, and microphone-output interfaces of identified language, text (translated) and audio (translated).

**Experimental Setup and Dataset**

The system's performance evaluation was done through the assembly of a controlled dataset composed of 200 audio samples (50 samples per language for Hindi, Spanish, French, and German). The samples were taken from the Mozilla Common Voice 13.0 dataset, while specific segments with different accents and speech speeds were selected.

**Test Conditions:**

- **Hardware:** The tests were run on a machine with an NVIDIA RTX 3060 GPU (12GB VRAM), 16GB RAM, and an Intel i7-11th Gen CPU.
- **Environment:** Evaluation was done under two conditions: Quiet (background noise <30 dB) and Noisy (simulated office environment noise at ~60 dB).
- **Baselines:** The proposed system (Whisper + MMS-TTS) was compared against a Standard Cascaded API Baseline (Google Speech-to-Text

+ Google Translate API + GTTS) to measure improvements in accuracy and latency.

## V. IMPLEMENTATION DETAILS

The system is written in Python, and it is based on machine learning and audio processing libraries.

**Technology Stack**

- Programming Language: Python 3.9+
- Machine Learning Frameworks: PyTorch [12] and Hugging Face Transformers [3].
- Audio Processing: Librosa [11], SoundFile [13], and NumPy [14].
- Web Interface: Gradio [5].
- Hardware: NVIDIA GPU for CUDA acceleration.

**Environment Setup**

Dependencies were handled by using a Python virtual environment. The CUDA was set to allow the use of the GPU hence greatly accelerating model inference in real time applications [15].

**Real-time Considerations**

In order to have a real-time experience, an optimization was done with model pre-loading, GPU acceleration, and effective VAD to filter out silent segments [16].

## VI. RESULTS AND DISCUSSION

The synthesis system was thoroughly tested to assess the linguistic accuracy of its translations and the overall quality of the generated English text. Additionally, the performance was measured by looking at the total time needed for synthesis to ensure computational efficiency.

**Performance Evaluation and Comparative Analysis**

The system was evaluated based on three main metrics: Word Error Rate (WER) for transcription accuracy, BLEU Score for translation quality, and End-to-End Latency for real-time performance. We evaluated our proposed MVP architecture based on local inference only (Whisper + MMS-TTS) against a cloud-based cascading baseline (Google STT + GoogleTranslate + GTTS).

TABLE I TRANSLATION PERFORMANCE

| Language | System | WER (%) ↓ | BLEU Score ↑ | Latency (s) ↓ |
|---|---|---|---|---|
| Hindi Spanish | Proposed(Whisper+MMS) | 11.2 | 38.4 | 2.4 |
| German | Baseline(Google API) | 15.6 | 32.1 | 3.1 |
| Average | Proposed(Whisper+MMS) | 8.4 | 44.2 | 2.6 |
| | Baseline(Google API) | 9.1 | 42.5 | 2.9 |
| | Proposed(Whisper+MMS) | 7.9 | 46.8 | 2.1 |
| | Baseline(Google API) | 8.5 | 44.0 | 2.5 |
| | Proposed(Whisper+MMS) | 9.1 | 43.1 | 2.37 |
| | Baseline(Google API) | 11.0 | 39.5 | 2.83 |

As shown in Table I, the suggested system achieved the best performance in all languages compared to the baseline. In Hindi, a language that is notoriously difficult for standard APIs, our system outperformed by 28.2% reduction in WER and thus had a very high BLEU score. Also, the local GPU acceleration gave an average response time of 2.37 seconds, around 16% faster than the cloud-based baseline.

**Discussion**

The findings put emphasis on the strength of the OpenAI Whisper model to work with various languages when it comes to speech recognition and the process of direct translation [1]. Association of MMS-TTS also makes sure that the translated text is translated into effective and decipherable voice.

Real time performance is an important consideration of the interactive communication of the system. The Gradio interface was found to be very useful in fast deployment and easy interaction by users [5]. The system is effective, but it may be improved in the following aspects:

- **Robustness to Noise:** Use of superior noise suppression
- **Latency Optimization:** Considering the smaller and more efficient models of the quick response.
- **Speaker Diarization**: Differences between more than one speaker.

## VII. CONCLUSION

The current paper has detailed out the design and the analysis of a real time speech-to-speech translation system. The system has a bridging effect on the language barrier by combining the OpenAI Whisper with MMS-TTS of Facebook. The architecture guarantees effective and accurate functionality by the means of strong preprocessing.

The experimental findings indicated that the accuracy of translation and the tolerated latency of the real-time in several languages were high. The system offers a convenient solution to translating languages on-the-fly, through a web interface where the transformative nature of the current state of AI can be revealed in terms of improving global interaction.

## VIII. FUTURE SCOPE

The future development may be aimed at incorporating more advanced VAD algorithms, experimenting with fined-tuning edge device models, and multi-target language support beyond the English language. Besides, it can be improved by experimenting with such features as speaker diarization when having a multi-party conversation.

## REFERENCES

1. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and M. Sutskever, "Robust Speech Recognition via Large-Scale Weak Supervision", arXiv preprint arXiv:2212.04356, 2023.
2. C. Wang, W. Chen, Y. Li, C. Chen, Y. Liu, Z. Qin, Y. Xia, K. Shrivastava, R. He, C. Chen, Et al., "Large-Scale Self-Supervised Pre-training for Multilingual Speech-to-Speech Translation", arXiv preprint arXiv:2303.04169, 2023.
3. T. Wolf, L. Debut, S. Sanh, J. Chaumond, C. Wolf, D. Vidal, M. Chaffin, H. Debut, V. von Platen, and D. J. F. L. et al., "HuggingFace's Trans-formers: State-of-the-art Natural Language Processing", Proc. EMNLP: System Demonstrations, 2019, pp. 41-46.
4. D. Jurafsky and J. H. Martin, Speech and Language Processing, 3rd ed., Pearson, 2021.
5. Abid, A. Khan, K. Popovic, and S. Mahmood, "Gradio: Hassle-free sharing of machine learning models in your browser", Proc. Machine Learning Research, vol. 147, pp. 1-6, 2021.
6. G. Hinton et al., "Deep Neural Networks for Acoustic Modeling in Speech Recognition", IEEE Signal Processing Magazine, vol. 29, no. 6, pp. 82-99, 2012.
7. van den Oord et al., "WaveNet: A Generative Model for Raw Audio", arXiv preprint arXiv:1609.03499, 2016.
8. Vaswani et al., "Attention Is All You Need", Advances in Neural Information Processing Systems, vol. 30, pp. 5998-6008, 2017.
9. S. Newman, Building Microservices: Designing Fine-Grained Systems, O'Reilly Media, 2015.
10. European Commission, Ethics Guidelines for Trustworthy Artificial Intelligence, Brussels, Belgium, 2019.
11. McFee et al., "librosa: Audio and Music Signal Analysis in Python", Proc. 14th Python in Science Conference, 2015.
12. Paszke et al., "PyTorch: An Imperative Style, High-Performance Deep Learning Library", Advances in Neural Information Processing Systems, vol. 32, 2019.
13. Beffa, "Python-soundfile: An audio library based on libsndfile", Journal of Open Source Software, vol. 1, no. 4, p. 195, 2016.
14. R. Harris et al., "Array programming with NumPy", Nature, vol. 585, no. 7825, pp. 357-362, 2020.
15. NVIDIA, "CUDA C++ Programming Guide", NVIDIA Corporation, 2023.
16. ITU-T Recommendation G.729 Annex B, "A Silence Compression Scheme for G.729 Optimized for Mixed Voice/Data Applications", 1996.

17. S. Nakamura and K. Itou, "Statistical Speech-to-Speech Translation", Acoustical Science and Technology, vol. 27, no. 2, pp. 81-87, 2006.
18. L. R. Rabiner and B. H. Juang, Fundamentals of Speech Recognition, Prentice Hall, 1993.
19. P. Koehn, Statistical Machine Translation, Cambridge University Press, 2009.
20. P. Taylor, Text-to-Speech Synthesis, Cambridge University Press, 2009.
21. Y. Wang et al., "Tacotron: Towards End-to-End Speech Synthesis", Interspeech 2017, pp. 4006-4010, 2017.
22. T. Young, D. Hazarika, S. Poria, and E. Cambria, "Recent trends in deep learning-based natural language processing", IEEE Computational Intelligence Magazine, vol. 13, no. 3, pp. 55–75, 2018.
23. R. T. Fielding, "Architectural styles and the design of network-based software architectures", Ph.D. dissertation, Univ. of California, Irvine, CA, USA, 2000.
24. Schuller et al., "Speech emotion recognition: Two decades in a nutshell", IEEE Signal Processing Magazine, vol. 35, no.1, pp. 111–119, 2018.