STM JOURNALS

International Journal of
Computer Science Languages

Volume 2, Issue 1, 2024
January–June
DOI (Journal): 10.37591/IJCSL

https://journals.stmjournals.com/ijcsl

Research IJCSL

Real-time Voice-over Translation

Jagdish W. Bakal[1], Anushka Salunke[2,*], Punnyai Suryawanshi[2], Neha Upadhye[2]

Abstract

The real-time voice-over translation project stands as a pioneering endeavor poised to reshape the landscape of cross-cultural communication. In response to the escalating demand for fluid interaction amidst linguistic diversity, this initiative sets out to engineer a transformative solution. Its core objective lies in the seamless mediation of language barriers during real-time exchanges. Through the fusion of cutting-edge speech recognition and machine translation technologies, the project endeavors to fabricate a dynamic platform. This platform will possess the remarkable capability to instantaneously transmute spoken utterances into translated textual counterparts. By harnessing the power of sophisticated algorithms, it aims to capture, interpret, and swiftly convert diverse languages into comprehensible text. Subsequently, employing state-of-the-art text-to-speech synthesis, the translated text will be elegantly vocalized, restoring linguistic harmony in the user's preferred language. The real-time voice-over translation project harbors immense potential to redefine the boundaries of global interaction, fostering fluid dialogue across linguistic frontiers. Its envisioned impact extends beyond mere convenience, empowering individuals and enterprises alike to engage effortlessly and authentically across linguistic divides. As it navigates the forefront of technological innovation, this project embodies a beacon of inclusivity, heralding a future where language ceases to be a barrier but becomes a bridge to shared understanding and collaboration.

Keywords: Real-time voice-over Translation Project, language barriers, real-time communication, speech recognition, machine translation

INTRODUCTION

Real-time voice-over translation (RVOT) is a technology that enables real-time translation of spoken language into another language, while the speaker is still speaking. It is a crucial tool in today's globalized world, where communication across different languages is essential. With the rise of international business, travel, and cultural exchange, the need for efficient and accurate translation has become increasingly important.

***Author for Correspondence**
Anushka Salunke
E-mail: anushkass21hcompe@student.mes.ac.in

[1]Principal, Department of Computer Science and Engineering, Pillai HOC College of Engineering and Technology, Raigad, Maharashtra, India
[2]Student, Department of Computer Science and Engineering, Pillai HOC College of Engineering and Technology, Raigad, Maharashtra, India

Received Date: April 24, 2024
Accepted Date: April 30, 2024
Published Date: May 09, 2024

Citation: Jagdish W. Bakal, Anushka Salunke, Punnyai Suryawanshi, Neha Upadhye. Real-time Voice-over Translation. International Journal of Computer Science Languages. 2024; 2(1): 24–32p.

RVOT works by using advanced software and machine learning algorithms to analyze the spoken words and convert them into the target language in real time. The translated text is then spoken out loud by a computer-generated voice or a human interpreter. This technology has enabled seamless communication between people speaking different languages in various settings, such as business meetings, conferences, and international events.

One of the primary benefits of RVOT is its ability to break down language barriers and promote cross-cultural understanding. It allows individuals to communicate without any delay or interruption, making the conversation flow more naturally and effectively. This can be particularly helpful in

business negotiations, where accurate and timely communication is vital for successful deals. RVOT also plays a significant role in promoting cultural exchange and understanding, as it allows people from different backgrounds to communicate without any language barriers.

Another key advantage of RVOT is its efficiency. Unlike traditional methods of translation, which require a person to listen, interpret, and then speak, RVOT can translate in real-time without any delay. This saves both time and effort, making communication more efficient and productive. Furthermore, with the use of artificial intelligence and machine learning, RVOT can continuously improve its accuracy and speed, making it an increasingly valuable tool for businesses and individuals. Furthermore, RVOT relies heavily on the quality of its source material, and it may struggle to translate slang, idioms, and dialects accurately. Therefore, the role of human interpreters is still crucial in situations that require precise and nuanced translations.

Moreover, some critics argue that the use of RVOT may lead to the loss of traditional language learning and cultural immersion. With the convenience provided by RVOT, people may become reliant on this technology, neglecting the importance of learning a new language. This could have negative implications for cultural preservation and understanding in the long run.

LITERATURE SURVEY

By utilizing a unique cross-lingual voice conversion model that was trained on a polyglot speech corpus, this study presents a multilingual text-to-speech (TTS) system. The system uses a neural vocoder based on HiFiGAN to extract speaker-invariant features from a pre-trained model and produce speech that is specific to the target speaker. Utilizing a synthetic dataset created by the model's translation of speaker identities from other languages into English is the first step in training a single-speaker multilingual TTS system. Nonetheless, the methodology faces obstacles like restricted fine-tuning capacity for particular languages, supporting heterogeneous linguistic structures, guaranteeing translation quality control, handling elevated computational demands, possible error propagation, and tackling challenges related to domain-specific language adaptation.

According to studies, it is simple and efficient to apply multilingual end-to-end speech translation (E2E-ST) using a universal sequence-to-sequence paradigm. In contrast to conventional pipeline systems, this approach preserves memory and processing capacity by using auditory signals from the source language to reduce error transmission. This permits the translation of several language pairs. Both the decoding and training procedures are streamlined by the universal sequence-to-sequence structure. Nevertheless, a significant disadvantage of end-to-end voice translation is the possibility of higher latency and cost. This is mainly due to the use of deep neural networks and complicated input features, which need a significant amount of time and computer power. In addition, while creating word connections between words in the source speech and words in the target language can be advantageous in terms of decreasing computational complexity and latency, vocabulary manipulation can present difficulties when compared to other translation approaches.

The goal of recent developments in end-to-end voice translation has been to reduce the spread of errors from speech recognition to translation. In this field, techniques including two-pass decoding, multi-task learning, and data augmentation have become popular. However, problems still exist since deep neural networks and large input feature sets are expensive and have a significant latency. Furthermore, creating word linkages between target and source speech words makes vocabulary manipulation more difficult while cutting down on processing time and complexity. Training instability and slowness are prevalent problems in the field of generative adversarial networks (GANs), where two networks (generator and discriminator) compete. This problem is made worse by the requirement for large amounts of training data to achieve the best possible outcomes. Furthermore, 16000 Hz is the usual sample rate used in vocoders, where waveforms are generated by pre-trained models that receive spectrograms as input.

METHODOLOGY

The algorithm listens for spoken input, translates it into the desired language, and then speaks out the translated text. This allows for real-time speech translation and synthesis. This Python script is designed to perform speech recognition, translation, and TTS synthesis. The algorithm for the provided code can be broken down into the following steps:

1. *Initialization*
   i. Import necessary libraries (‘speech_recognition’, ‘translate’, ‘pyttsx3’).
   ii. Initialize the Speech Recognition (‘recognizer’) and Text-to-Speech (‘engine’) objects.
2. *Transcribe audio function (‘transcribe_audio(language)’)*
   i. Open the microphone as a source for audio input.
   ii. Adjust for ambient noise to improve recognition accuracy.
   iii. Listen to the audio input using the ‘listen()’ method.
   iv. Attempt to recognize the speech using Google's speech recognition service (‘recognize_google()’).
   v. Return the recognized text.
3. *Translate text function (‘translate_text(text, target_language)’)*
   i. Create a Translator object with the target language specified.
   ii. Translate the provided text using the ‘translate()’ method of the Translator object.
   iii. Return the translated text.
4. *Speak text function (‘speak_text(text, speak_language)’)*
   i. Initialize the Text-to-Speech engine.
   ii. Set the desired voice for speaking based on the specified language.
   iii. Speak the provided text using the ‘say()’ method.
   iv. Run and wait for the speech to complete using ‘runAndWait()’.
   v. Stop the engine to release resources using ‘stop()’.
5. *Main execution*
   i. Prompt the user to input the target language and the language they will be speaking in.
   ii. Continuously transcribe audio input using the specified language until the user says “stop”.
   iii. Translate the transcribed text into the target language.
   iv. Speak the translated text in the specified language.
6. *Termination*
   i. Exit the program when the user says “stop”.

Following is a breakdown of the methodology used in the code:

1. *Speech recognition*
   i. The ‘transcribe_audio()’ function utilizes the SpeechRecognition library to transcribe audio input from the microphone.
   ii. It first adjusts for ambient noise to improve recognition accuracy.
   iii. Then, it listens to the audio input and attempts to recognize speech using Google's speech recognition service (‘recognize_google’ method).
   iv. If speech is recognized, the transcribed text is returned.
2. *Translation*
   i. The ‘translate_text()’ function uses the ‘translate’ library to translate text from one language to another.
   ii. It takes the transcribed text and the target language as input parameters.
   iii. The Translator object is initialized with the target language, and the ‘translate()’ method translates the text.
   iv. The translated text is returned.
3. *Text-to-Speech*
   i. The ‘speak_text()’ function uses the pyttsx3 library to convert text to speech.
   ii. It initializes the Text-to-Speech engine (‘pyttsx3.init()’), sets the voice language, speaks the provided text, and then waits for the speech to finish (‘runAndWait()’).
   iii. Finally, it stops the engine to release resources (‘engine.stop()’).

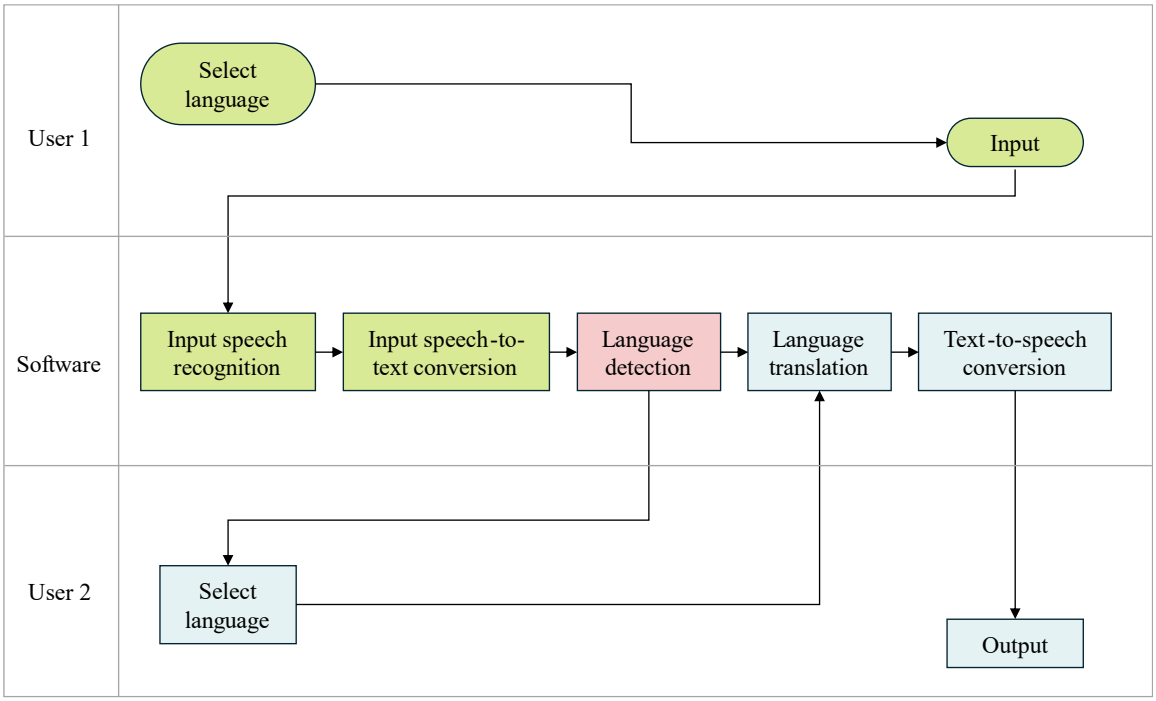

Figure 1. Swim lane diagram.

4. *Main execution*
   i. The script prompts the user to input the target language code (language to translate to) and the speak language code (language to speak in).
   ii. It then enters a loop where it continuously listens to the user's speech input.
   iii. If the user says "stop", the loop breaks and the program ends.
   iv. Otherwise, it transcribes the speech, translates it to the target language, prints the translated text, and speaks it in the specified language.

Swim lane diagrams are graphical representations that depict the progression of a process from its initiation to its completion. Additionally, these graphics illustrate the individuals or entities accountable for each individual stage inside the process.

Figure 1 depicts a software-driven procedure for converting spoken language between users. The system is partitioned into two distinct areas, namely User 1 and User 2, while a central "Software" segment is responsible for executing the translation process. User 1 provides a linguistic input, which is subsequently processed by the software.

The software uses speech recognition to convert spoken language into text, subsequently identifying the language and providing the user with the translated text. User 2 also chooses a language, and the translated speech is then returned to the user. The flowchart employs straight lines to link steps, and the design is sleek and uncomplicated, including a white backdrop and black text.

SYSTEM ARCHITECTURE

The flowchart (Figure 2) illustrates a trinary process for transforming spoken language into translated text and audio. The process commences with an input language box denoted as "Input Language" and a desired output language box labeled as "Preferred Output Language." The user's input is analyzed using a speech recognition library, which is then followed by the initial stage called "Speech to Text Conversion." In the second phase, the text is translated into the desired output language using the translation and TTS software libraries. In the third phase, the translated text is transformed into speech, resulting in the final output known as "Translated Text and Audio."

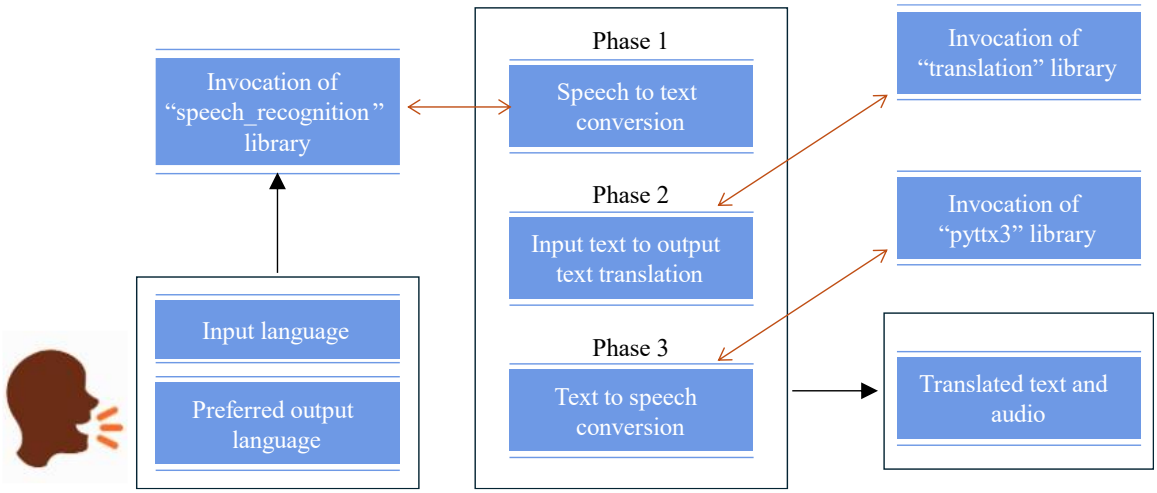

Figure 2. Proposed system architecture.

Working

RVOT technology is a cutting-edge system that allows for spoken conversation to occur seamlessly between individuals who speak different languages [1]. Advancements in technology have enabled the real-time translation of spoken words without any latency or interruption [2]. This facilitates communication between individuals who speak multiple languages in a simple and cost-effective manner. This technology is specifically developed to offer translation services without requiring the presence of a local interpreter, hence increasing its accessibility and availability to anyone worldwide. The RVOT technique incorporates the conversion of spoken language into text for translation, followed by the utilization of AI and machine learning algorithms to translate the text into another language [1]. After being translated, the text is subsequently transformed into voice that sounds natural utilizing techniques for TTS synthesis. Modern neural machine translation (NMT) systems have the ability to facilitate communication between individuals who speak different languages. They achieve this by analyzing unique waveforms produced by a word, phrase, or sentence to determine the original language and comprehend the speaker's intended message. Subsequently, they generate a translation in the desired language [4]. The use of real-time voice over translation technology has the capacity to revolutionize global communication by reducing delays and accuracy problems, eliminating the necessity for manual dubbing or employing voice actors, and delivering consistent and high-quality voice overs for videos in various languages. The text consists of the numbers 4, 3, and 1 enclosed in square brackets.

TOOLS REQUIREMENT

The code snippet uses several software tools and libraries to transcribe and translate speech in real-time. It also relies on specific hardware components to capture and process audio [5]. Following is a detailed description of the tools used:

1. *Software tools*
   i. *Python:* Python is a programming language that is interpreted and operates at a high level. It is commonly used for general-purpose programming. The code sample predominantly utilizes the major language.
   ii. *Speech recognition:* A number of voice recognition engines, such as Google Voice Recognition, are accessible through the SpeechRecognition Python package. It enables the software to convert spoken language into text [6].
   iii. *Translate:* The Translate Python package offers an easy-to-use interface for connecting to different translation services and translating text. To convert the text from transcription to the target language, this code is utilized [7].
   iv. *pyttsx3:* One Python library that can convert text to speech is pyttsx3. It offers a straightforward interface for turning text into speech and supports numerous TTS engines.

2. *Hardware components*
   i. *Microphone:* A microphone is utilized to record user-input audio. The voice recognition engine cannot function without the ability to transcribe spoken language [8].
   ii. *Speaker:* Audio output of the translated text is achieved by the use of a speaker. The pyttsx3 package is utilized to transform the translated text into audible version.
3. *Operating system*
   i. This code is cross-platform, meaning it can execute on any operating system that has Python and the necessary libraries installed. It is cross-platform, meaning it should run well on Linux, macOS, and Windows.
4. *Internet connection*
   i. Google Speech Recognition and the translation service both need an active internet connection to transcribe audio and translate text, respectively. Having access to the internet is crucial for these services to work [9, 10].

RESULT ANALYSIS

Figure 3 depicts the project as a speech translation program with a GUI (graphical user interface) that allows users to input speech in one language, translate it to another, and hear the translated text spoken aloud, based on the code and assumptions. Key features and components are summarized here. Detect Speech Using the 'speech_recognition' library to transcribe microphone input. Use the 'translate' library to convert transcribed text to the target language. Text-to-Speech calls the 'pyttsx3' library to speak the translated text aloud. A simple GUI is created using 'tkinter' to allow user interaction, including input areas for target and speak languages, buttons for translation start/stop, and a display area for translated text. Collected Data For analysis and visualisation, stores timings, accuracy, and lag times. Graphing and analysis: The ability to graph speech translation accuracy and lag time is included. Real-time speech translation is the project's goal to help users communicate in multiple languages. Its performance and accuracy depend on the voice recognition and translation services, the user's environment, and the application's implementation.

Figure 4 depicts the accuracy breakdown as a pie chart as total, SR, and Trans. SR stands for speech recognition, and Trans for translation accuracies. In terms of accuracy, the Trans provides 50%, the Overall provides 43.8%, and SR provides 6.3%. Figure 5 depicts THE Lag time of the project in the form of line graph. The lag time fluctuates from 2 to 8 seconds. Thus, we can conclude that the average lag time of project is approximately 5 or 5.5 seconds.

In order to evaluate the outcomes of your project, we can examine various facets derived from the given code and assumptions:

- The precision of the speech recognition and translation process has significant importance. While you have suggested utilizing a placeholder value of 80% for accuracy, it is necessary to empirically evaluate this in a real-world context by employing a test dataset. The lag time, which refers to the duration between the input of speech and the output of translation, is a significant statistic. The implemented graph illustrates the temporal variation of lag time, facilitating the identification of patterns or concerns.
- The user interface and the flow of interaction have the potential to influence the overall user experience. Evaluating the level of intuitiveness and user-friendliness of the interface might yield valuable insights regarding prospective enhancements. The code incorporates error handling mechanisms to address instances of voice recognition failure. The assessment of error handling efficacy and its influence on the holistic user experience holds significant importance.
- The usability of the program can be influenced by its performance, encompassing factors such as speed and resource utilization.
- The implementation of a feedback mechanism to collect user comments regarding the quality of translation and overall user experience might yield significant insights for subsequent iterations. The analysis can also include an evaluation of the languages that your application supports and their corresponding performance.

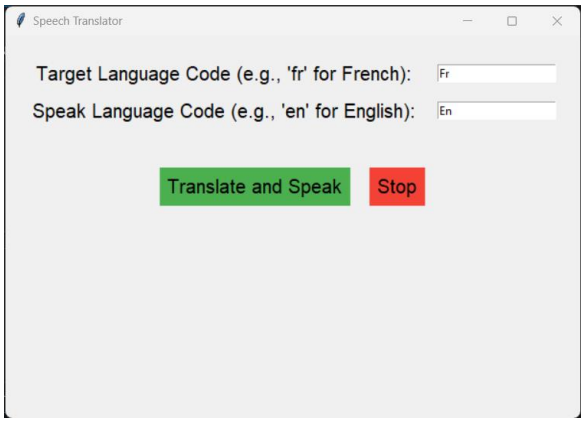

Figure 3. Interface.

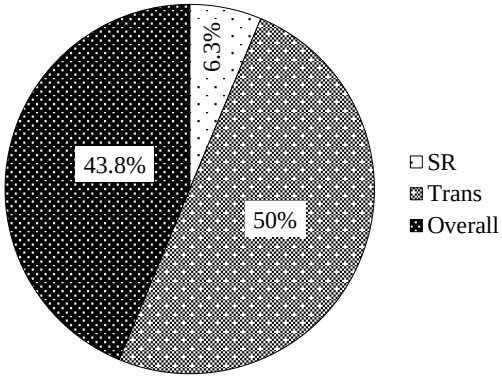

Figure 4. Accuracy pie chart.

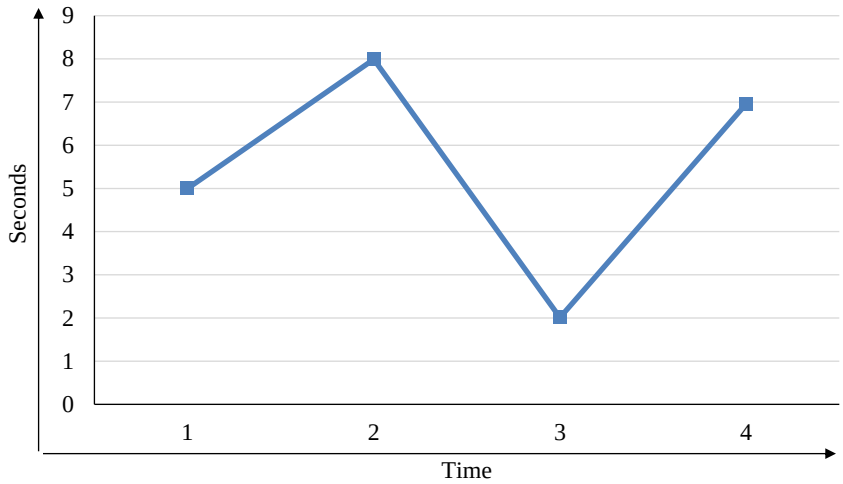

Figure 5. Lag time line graph.

In summary, a thorough examination of your project should encompass these elements and possibly other ones, contingent upon the particular objectives and prerequisites of the project.

CONCLUSION

To summarize, our study in this paper highlights the significant impact that creative solutions can have on our overall achievements. The real-time voice-over translation project represents a notable advancement towards a more integrated and inclusive global community. The project has the potential to revolutionize global communication by breaking down language barriers and fostering collaboration.

Limitations

- *Language support:* The code supports translation to a target language and speech recognition from a source language based on user input.
- *User interface:* The code lacks a user-friendly interface, which may limit its accessibility and usability for non-technical users.
- *Error handling:* The code provides basic error handling for speech recognition and translation services.
- *Time constraints:* Given our status as bachelor's students, it is a significant concern, to juggle numerous tasks simultaneously.

Future Scope

As we look to the future, there are several important areas where the real-time voice-over translation project may be improved and developed further. Firstly, studying the possible implementation of bidirectional mode could greatly increase the system's usefulness by permitting smooth language-based communication between users. Furthermore, the field of audio processing could improve by supporting audio files in addition to microphone input, hence requiring preprocessing steps to maximize data accuracy and quality. To increase the system's utility and reach, there are also a lot of opportunities for integration with already-existing platforms and applications, such communication tools. By taking use of these chances, the project can develop further into a flexible and essential instrument for promoting international communication and removing linguistic barriers.

Acknowledgements

We sincerely thank Dr. J.W. Bakal for his essential advice and consistent assistance during this study. Dr. Bakal's knowledge, guidance, and support have been crucial in determining the course of our work and helping us achieve success. His ability to navigate the difficulties of the project has been greatly aided by his insightful input, patience, and dedication. His unwavering commitment to quality has also motivated us to pursue the highest standards. We are incredibly appreciative of Dr. Bakal's mentorship, which has enhanced our academic experience and made significant contributions to our professional and personal development. We are grateful for your constant advice and faith in our abilities, Dr. Bakal.

REFERENCES

1. Wu J, Polyak A, Taigman Y, Fong J, Agrawal P, He Q. Multilingual text-to-speech training using cross language voice conversion and self-supervised learning of speech representations. In: 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Singapore, May 23–27, 2022. pp. 8017–8021.
2. Gupta AK, Somkunwar R, Kumari A, Kumari A, Godhke K, Repale J. Web based multilingual real time speech transcription transliteration and translation system. In: 2021 5th International Conference on Information Systems and Computer Networks (ISCON), Mathura, India, October 22–23, 2021. pp. 1–5).
3. Chaudhari S, Shukla A, Gaware T, Shinde N. Real time direct speech-to-speech translation. Int J Adv Res Sci Commun Technol. 2022; 2 (3): 881–884.
4. Patel D, Kudalkar M, Gupta S, Pawar R. Real-time text & speech translation using sequence to sequence approach. In: 2021 Third International Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore, India, September 2–4, 2021. pp. 722–727.
5. Bohouta G, Këpuska VZ. Real Time Speech Translation (Google TV and Chat). [Online]. ResearchGate. January 2018. Available at https://www.researchgate.net/publication/322686710_Real_Time_Speech_Translation_Google_TV_and_Chat
6. Ping L. Mobile platform speech intelligent recognition and translation app. In 2021 13th International Conference on Measuring Technology and Mechatronics Automation (ICMTMA), Beihai, China, January 16–17, 2021, pp. 595–598.
7. Inaguma H, Duh K, Kawahara T, Watanabe S. Multilingual end-to-end speech translation. In 2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU), Singapore, December 14–18, 2019. pp. 570–577.

8. Kim JW, Yoon H, Jung HY. Linguistic-coupled age-to-age voice translation to improve speech recognition performance in real environments. IEEE Access. 2021; 9: 136476–136486.
9. Krupakar H, Rajvel K, Bharathi B, Deborah SA, Krishnamurthy V. A survey of voice translation methodologies—Acoustic Dialect Decoder. In 2016 International Conference on Information Communication and Embedded Systems (ICICES), Chennai, India, February 25–26, 2016. pp. 1–9.
10. Franco E, Matamala A, Orero P. Voice-Over Translation: An Overview. Bern, Switzerland: Peter Lang; 2010.

