Voxify: Your Gateway to Multilingual Conversations

Prof. Krishnendu Nair[1], Siddhi Thoke[2], T T K Urshitha Sai[3], Pratham Yadav[4]

[1,2,3,4] *Department of Computer Engineering Pillai College of Engineering, New Panvel*

***Abstract*—This paper presents the development of a speech-to-speech translation system built on a hybrid architecture that integrates on-device processing with cloud-based services. The system comprises three core modules: speech-to-text transcription, translation and speech synthesis, and language detection, working in tandem to convert spoken input into translated audio output. Initial speech is captured using the device's microphone, with preprocessing techniques like noise reduction applied for clarity. On- device speech recognition ensures rapid transcription of spoken words into text, minimizing latency. The transcribed text is then sent to the cloud-based Bhashini API, which performs both text-to-text translation and text-to-speech synthesis. The system uses the Whisper speech-to-text model, fine-tuned for Indic languages, to detect the spoken language and ensure accurate translation. Error-handling mechanisms, data privacy protocols, and performance optimizations, such as compression techniques and encrypted communication, enhance the system's robustness. By combining fast on-device transcription with advanced cloud-based translation, the system delivers scalable, real-time translations, particularly suited for multilingual and culturally diverse contexts.**

***Index Terms*—Machine Translation, Natural Language Processing, Automatic Speech Recognition, Speech to Text Translation, Text-to-text translation, Text-to-Speech Synthesis, Speech Detection.**

I. INTRODUCTION

1.1 Fundamentals

Language barriers hinder communication across cultures [1], limiting travel experiences, business interactions, and educational opportunities. Traditional methods like dictionaries or human translators have drawbacks, but advancements in speech recognition, machine translation, and text-to-speech technologies offer promising solutions [2,6]. This research leverages these advancements to propose a mobile speech-to-speech translation application that tackles dialect and accent variations, promoting real-time communication across languages [7,3]. This user-friendly application contributes to human-computer interaction and machine translation by fostering inclusivity and global understanding in various domains [8].

1.2 Objectives

The ever-growing tapestry of human interaction is constantly challenged by language barriers. This research delves into a novel speech-to-speech translation application that leverages the power of the AI 4 Bharat model to break down these barriers and foster seamless communication [1]. Our application tackles the complexities of spoken language in a three-pronged approach:

Automatic Speech Recognition (ASR) acts as the initial bridge, transforming the acoustic properties of speech (pitch, frequency) into digital text. However, challenges like background noise and diverse speaking styles can introduce errors at this stage. Machine Translation (MT) then takes the baton, translating the recognized text across languages. Here, we explore two primary approaches: Statistical MT, which identifies patterns between languages, and Neural MT, which utilizes deep learning for more natural- sounding translations. However, both grapple with limitations in handling idiomatic expressions and the intricate nuances of human language. Finally,

Text-to-speech synthesis (TTS) transforms the translated text back into speech, presenting the challenge of replicating natural human inflections [2]. This research goes beyond simply introducing the application. We will delve deeper into the functionalities and inherent limitations of these core technologies (ASR, MT, TTS) within the context of speech-to-speech translation. Furthermore, we will analyze how the AI 4 Bharat model, with its diverse data training regimen, addresses these limitations.

Specifically, we will focus on how it enhances accuracy and robustness in handling various dialects and accents,
1
ultimately leading to superior speech-to-speech translation [7,3]. This research aspires to not only showcase the application's contribution to the field of Natural Language Processing (NLP) translation but also pave the way for the development of more user-friendly and sophisticated language translation tools that can empower truly global communication [8].

1.3 Scope

This project centers on developing a speech-to-speech translation application powered by the AI 4 Bharat model. The core deliverable is a functional application that automatically detects spoken language, allows users to choose their preferred translation language, and integrates ASR, MT (utilizing AI 4 Bharat), and TTS functionalities [1,2]. We will focus on core functionalities, user interface, and ensuring the application is robust across various speech patterns. The target audience is broad, encompassing anyone seeking to bridge language barriers in spoken communication [6]. Project success hinges on delivering an accurate, user-friendly application that leverages AI 4 Bharat's strengths and is well-documented for both users and future developers [3].

1.4 Outline

This report outlines the development of a speech-to-speech translation application powered by the AI 4 Bharat model [1]. The first chapter, Introduction, will introduce the motivation for tackling language barriers and provide a project overview highlighting the application's functionalities [6]. It will then clearly outline the research objectives, such as exploring core technologies and showcasing the advantages of AI 4 Bharat [2]t. Chapter 2, Proposed System, will delve into the Architecture: Existing System Architecture and Proposed System Architecture. Here, the aim is to outline the proposed system architecture for the speech-to-speech translation application. It provides a detailed description of how the application operates, highlighting its core components and functionalities. This chapter will also outline the technical requirements and methodology for implementing the proposed speech-to-speech translation application. Finally, the Conclusion in Chapter 4 will summarize the key findings, highlight the application's potential impact, and briefly mention areas for further development.

II. Literature Survey

Language barriers hinder effective communication globally. Speech-to-speech translation applications, utilizing components like ASR, MT, and TTS, offer promising solutions. However, challenges such as background noise and idiomatic expressions impede accuracy. ASR struggles with diverse accents, while MT faces difficulties in handling language nuances. TTS synthesis often produces robotic-sounding output. The AI 4 Bharat model aims to address these limitations by enhancing accuracy and robustness, particularly in diverse dialects and accents (Grosjean, 2010; Levin et al., 2020; Hirschberg & Manning, 2015; Koehn, 2009; Taylor & Black, 1998; Narayan et al., 2022).

2.1 Literature Review

Paper 1**:**
Jay Gala, Pranjal A. Chitale, Raghavan AK, Varun Gumma, “IndicTrans2: Towards High-Quality and Accessible Machine
Translation Models for all 22 Scheduled Indian Languages” (Dec 2023). Towards High-Quality and Accessible Machine Translation Models for all 22 Scheduled Indian Languages" addresses the deficiency in robust machine translation systems for India's diverse linguistic landscape. It introduces the Bharat Parallel Corpus Collection (BPCC) and presents IndicTrans2, the first translation model supporting all 22 languages. Emphasizing the significance of accessible and high-quality MT for social inclusion and national integrity, the paper showcases notable advancements in neural machine translation for Indian languages [10].

Paper 2:
Abhinav Jha, Sumit Kumar Jindal, Hemprasad Yashwant Patil, Sardar M N Islam, “Multilingual Indian Language Neural Machine Translation System Using mT5 Transformer.” (2023)[11]. The assessment of a multilingual neural machine translation system designed for Indian languages.

Built upon the mT5 transformer architecture, a variant of the T5 transformer known for its success in developing cutting-edge NLP models, the system was trained using the modified Asian Language Treebank multilingual dataset. This training aimed to enable the system to proficiently translate text between English, Hindi, and Bengali. Impressively, the system yielded satisfactory BLEU scores, surpassing 20 in five out of six language pairs. Notably, the English-to-Bengali translation system achieved a remarkable maximum BLEU score of 49.87, while the Bengali-to-English translation system attained an average BLEU score of 42.43 (Jha et al., 2023) [11].

Paper 3:

Aarati H. Patil; Snehal S. Patil; Shubham M. Patil; Tatwadarshi P. Nagarhalli, "Real Time Machine Translation System between Indian Languages." (2022). A recent study addresses the challenge of linguistic diversity in India through the development of a deep learning-based translation system. Utilizing Long Short- Term Memory (LSTM) models, renowned for their ability to process long sequences of data, the system demonstrates effective multilingual translation capabilities. However, challenges such as resource-intensive training, potential quality variations based on data quantity and quality, and limitations in handling complex grammar rules and lengthy sequences must be considered for real- world deployment (Patil et al., 2022) [12].

Paper 4:

B.S Harish, R. Kasturi Rangan, "A comprehensive survey on Indian regional language processing" (2022). The paper provides a comprehensive survey on Indian regional language processing, addressing the challenges posed by the multilingual nature of internet content. It reviews various approaches and techniques employed by researchers in tasks such as machine translation, Named Entity Recognition, Sentiment Analysis, and Parts- OfSpeech tagging, encompassing Rule, Statistical, and Neural methods. Additionally, the paper discusses the challenges motivating language processing solutions, outlines available datasets for Indian regional languages, and explores future prospects and requirements for enhancing language processing in this context (Harish & Rangan, 2022) [13].

Paper 5:

Study of Regional Language Translator Using Natural Language Processing. Authors: P.Santhi, J.Aarthi, S.Bhavatharini, N.Guna Nandhini & R.Snegha. The study by P.Santhi et al. explores the development of a regional language translator using Natural Language Processing (NLP) techniques. The proposed method aims to enable people from various regions to easily understand the essence of government drafts, bills, and amendments, which are often published only in Hindi and English. By leveraging NLP tools, the system processes the input text and translates it into the user's preferred regional language. This work represents a valuable contribution towards improving accessibility and comprehension of important government communication, particularly for those who may not be proficient in the official languages. The development of such translation systems can foster greater inclusion and understanding among diverse communities.[14]

Paper 6:

Improving English-to-Indian Language Neural Machine Translation Systems. Authors:Akshara Kandimalla , Pintu Lohar , Souvik Kumar Maji, Andy Way. The authors conclude that back-translation can be a useful technique to improve the performance of English-to-Indian language Neural Machine Translation (NMT) systems, but it is more helpful for weaker baseline models. They also found that BLEU scores may not always accurately reflect the quality of machine translation. In future work, the authors plan to explore the use of monolingual datasets of various sizes and domains to determine the effectiveness of back-translation. [15]

Paper 7:

Experience of neural machine translation between Indian Languages. Authors: Shubham Dewangan, Shreya Alva, Nitish Joshi, Pushpak Bhattacharyya. The authors conclude that BPE is a useful technique for NMT, but the optimal number of merge operations can vary depending on the language pair. They found that a smaller number of merge operations is generally better for Indian languages. [16]

Paper 8:

Natural Language Processing based Machine

Translation for Hindi-English using GRU and AttentionAuthors: Jaskirat Singh; Sansriti Sharma; Briskilal J. The document concludes that there is a lot of research going on to improve the ways computers can process Indian regional languages. This is because these languages are complex and have many different dialects. However, there has not been as much research done on this topic compared to other languages. [17]

Paper 9:

Natural language processing: state of the art, current trends and challenges. Authors: Aditya Koli, Kiran Khatter, Sukhdev Singh. The document concludes by discussing future directions in NLP, including dialogue systems and applications in medicine. NLP is expected to be used in future systems that can enable robots to interact with humans in natural languages. In the field of medicine, NLP is being used to develop systems that can extract and summarize information from patient records. [18]

Paper 10:

Natural Language Processing and Its Applications in Machine Translation. Authors: A Diachronic Review. Kai Jiang, Xi Lu. The article emphasizes the need for additional research to assess the potential benefits and drawbacks of artificial intelligence (AI) in translation studies. Stakeholders have differing opinions on the impact of AI on translation quality. Some experts believe that AI can improve translation precision and efficiency, while others are skeptical about the ability of AI to completely replace human translators. [19]

III. PROPOSED SYSTEM

3.1 Overview

This research proposes a novel speech-to-speech translation application designed to bridge communication gaps across diverse languages [1]. The application leverages the power of the AI 4 Bharat model, known for its robust performance in handling various dialects and accents [2]. The core functionality begins with Automatic Speech Recognition (ASR), which acts as the ears of the application [7]. ASR tackles the challenge of converting spoken language, with its complexities of pitch and frequency, into digital text. The application then automatically detects the language of the spoken input [3]. Next, Machine Translation (MT) takes center stage. Here, the application harnesses the strengths of the AI 4 Bharat model [1]. Unlike traditional MT models, AI 4 Bharat is specifically trained on a vast dataset encompassing diverse languages, dialects, and accents. This unique training approach empowers the application to deliver superior translation accuracy, even when faced with regional variations in speech patterns [7]. Finally, Text-to-Speech Synthesis (TTS) transforms the translated text back into naturalsounding speech in the user's preferred language [8]. This entire process, from capturing spoken language to delivering the translated speech unfolds seamlessly within the application, fostering real-time communication across language barriers.

3.1.1 Existing System Architecture

The existing system implements a text-to-text translation architecture centered on the Transformer encoder-decoder framework. This web-based application functions within a predefined set of languages, typically around 21. Users interact with the system by providing text input in their source language. The system then leverages the Transformer model to translate the input text into the user-chosen target language. The translated text forms the core output, displayed within the web interface for user review. This architecture, while functional, presents limitations in its inability to handle spoken language input and its restriction to text-based communication.

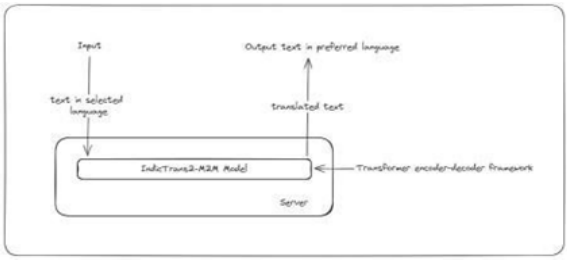

Input
text in selected language
Output text in preferred language
translated text
IndicTrans2-M2M Model
Transformer encoder-decoder framework
Server

Fig 3.1 Existing System Architecture

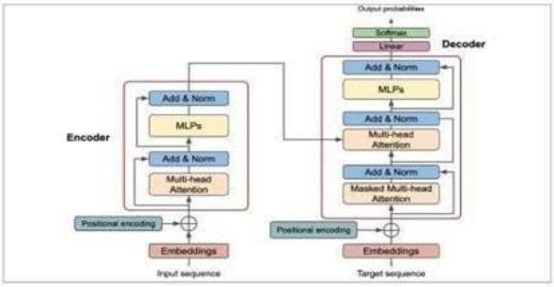


Fig 3.2 Transformer encoder-decoder framework

3.1.2 Proposed System Architecture

This research proposes a novel speech-to-speech translation application designed for seamless communication across diverse languages [1]. The architecture prioritizes a user- centric mobile application experience while leveraging robust server-side functionalities for accurate translation [2].

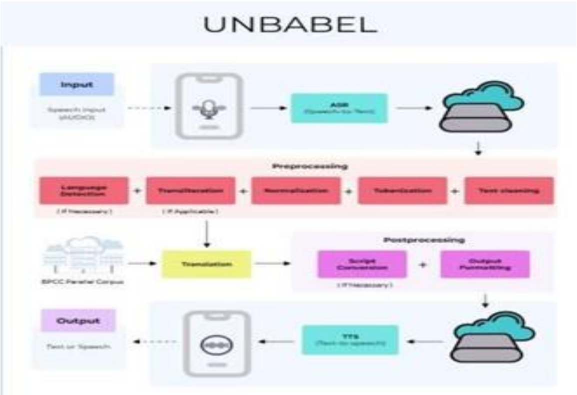


Fig 3.3 Proposed System Architecture

Mobile Application:

•Speech Recognition Module: This module utilizes Automatic Speech Recognition (ASR) technology to convert the user's spoken language (including dialects and accents) into digital text.

•User Interface: Provides functionalities for voice input, language selection (source and target), and playback of the translated speech.

•Text-to-Speech (TTS) Engine: Converts the translated text back into natural-sounding speech in the user's chosen target language (integrated within the mobile app).

•Secure Communication Channel: Ensures secure transmission of the user's speech data (converted text) to the server for processing.

Server-Side Development:

•Language Detection: Analyses the converted text to identify the source language automatically.

•Machine Translation Model: Integrates a powerful Machine Translation (MT) model, such as IndicTrans2, specifically trained for accurate translation between Indian languages.

3.2 Requirement for Implementation

Mobile App Development:

- Cross-Platform Development Framework**:** A suitable framework like React Native or Flutter will be chosen to ensure the application functions seamlessly on both iOS and Android devices [6].
- Speech Recognition Module Integration: A robust Automatic Speech Recognition (ASR) library or SDK, such as Google Speech-to-Text or Vosk, will be integrated to handle the complexities of diverse dialects and accents within spoken language [2].
- User-Centric Interface Design: The user interface will be designed with usability in mind, providing intuitive functionalities for voice input, source and target language selection, and clear playback of translated speech [7].
- Text-to-Speech Engine Integration**:** A high-quality Text-to-Speech (TTS) library or SDK, such as Google Text- toSpeech or iSpeech, will be integrated to deliver natural- sounding speech in various target languages, enhancing the user experience.

- Secure Communication Protocol: Secure communication protocols like HTTPS will be implemented to safeguard the privacy and confidentiality of user data during transmission between the mobile app and the server.

Server-Side Development:

- Server-Side Programming Language: A server-side programming language like Python or Java will be chosen based on its suitability for efficient data processing and API integration.
- Machine Translation Model Integration: The chosen Machine Translation (MT) model, such as IndicTrans2, will be integrated through its respective API or by deploying the model directly on the server [1].
- Language Detection Library: A language detection library like langdetect will be integrated to accurately identify the source language of the converted text received from the mobile app [3].
- API Development: APIs will be developed to facilitate communication between the mobile

app and the server for data exchange, enabling the transfer of converted text and the return of translated text.

3.2.1 Implementation

The implementation of the speech-to-speech translation system is built on a hybrid architecture that combines on- device and cloud-based processing to provide efficient and accurate translations. The system primarily consists of three modules: speech-to-text transcription, translation and speech synthesis, and language detection. These modules work together to convert spoken input into translated audio output.

Initially, audio input is captured using the device's microphone, with preprocessing techniques such as noise reduction applied to enhance clarity. The system uses the device's native speech recognition service to transcribe spoken words into text. This on-device transcription minimizes latency, enabling a rapid response and providing an initial text output of the user's speech. The transcribed text is then sent to the Bhashini API, a cloud-based service that performs both text translation and text-to-speech synthesis. The API generates the translated text in the desired target language and synthesizes it into audio, which is subsequently played back to the user.

For language detection, the system employs the Whisper speech-to-text model, which has been fine-tuned for multilingual transcription, particularly focusing on Indic

languages. This model processes the audio input and generates

a textual output. The transcribed text is then analyzed using a language detection algorithm that identifies the language spoken by the user. This detected language is used to set the appropriate source language for the Bhashini API, ensuring the translation is accurate and contextually appropriate.

Throughout the process, error-handling mechanisms ensure robust performance, addressing issues such as network failures or transcription inaccuracies. The system prioritizes user privacy by encrypting data transmission and limiting data storage. The combination of on-device speed with the advanced capabilities of cloud services results in a scalable and adaptable solution for real-time speech-to-speech translation, making it suitable for various linguistic and cultural contexts.

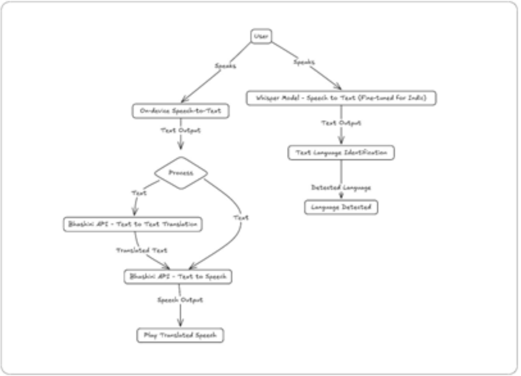


Figure 1.2 High-Level Architecture of Unbabel [1]

The proposed speech-to-speech translation system employs a hybrid architecture that integrates both on-device processing and cloud-based services to achieve efficient and accurate multilingual translation. The methodology involves several stages, ensuring seamless transcription, translation, and synthesis of speech, with a focus on supporting various Indic languages. Below is an overview of the key components and processes involved in the implementation.

The system begins with the input capture and pre-processing phase, where the user's speech is recorded using the device's microphone and stored in a compatible audio format, such as

.wav. To ensure the clarity of the speech input, noise reduction techniques are applied, and audio levels are normalized before further processing. This step ensures that the quality of the audio is consistent, improving the performance of subsequent transcription and translation processes.

In the speech-to-speech translation module, the first step is on- device speech-to-text transcription. This is achieved using the native speech recognition capabilities of the device, such as the Google Speech API on Android or Apple's Speech framework on iOS. The audio input is processed directly on the device, converting spoken words into a textual representation. This local processing reduces latency as it avoids the need to transmit audio data over the network, allowing for quick responses in the initial phase of the translation. Once the text transcription is generated, it is validated for accuracy, with mechanisms to prompt the user for a repeat or retry in case of errors or incomplete transcription. The output at this stage is a text string that accurately represents the user’s spoken words.

Following transcription, the text is sent to the Bhashini API for translation and speech synthesis. The text is transmitted securely to the API along with the source and target language specifications. The API then performs text-to-text translation, converting the text from the source language to the desired target language. The translation process utilizes pre-trained models optimized for the linguistic characteristics of Indic languages, ensuring accurate and culturally appropriate translations. The translated text is subsequently converted into speech using the API’s text-to-speech (TTS) synthesis capabilities, which generate an audio output that preserves the pronunciation and intonation of the target language. The resulting speech is then processed for playback, ensuring that it is delivered with clarity to the user through the device’s speakers. The combination of on- device transcription and cloud-based translation and synthesis allows the system to maintain low latency while benefiting from advanced translation capabilities.

In parallel, the system utilizes a language detection module to determine the language of the spoken input. This module employs the Whisper speech-to-text model, a multilingual transcription model fine-tuned specifically for Indic languages. The Whisper model processes the user’s audio input, generating a text transcription that captures the nuances of regional pronunciations and dialects. The fine-tuning of the model enables it to accurately handle a diverse range of phonetic variations commonly encountered in spoken Indic languages. The output is a text string representing the spoken input, which is then used for language identification.

The text-based language identification component processes the transcribed text using custom language detection algorithms or libraries like langdetect or fastText. These algorithms are trained on extensive datasets containing text samples from various Indic languages. The language detection process involves analyzing the linguistic features of the transcribed text and comparing them against known patterns for different languages, including code- mixed texts common in Hinglish and other regional variations. The output is a label that indicates the most likely language of the user’s spoken input, which helps ensure that the correct source language is selected for the translation process.

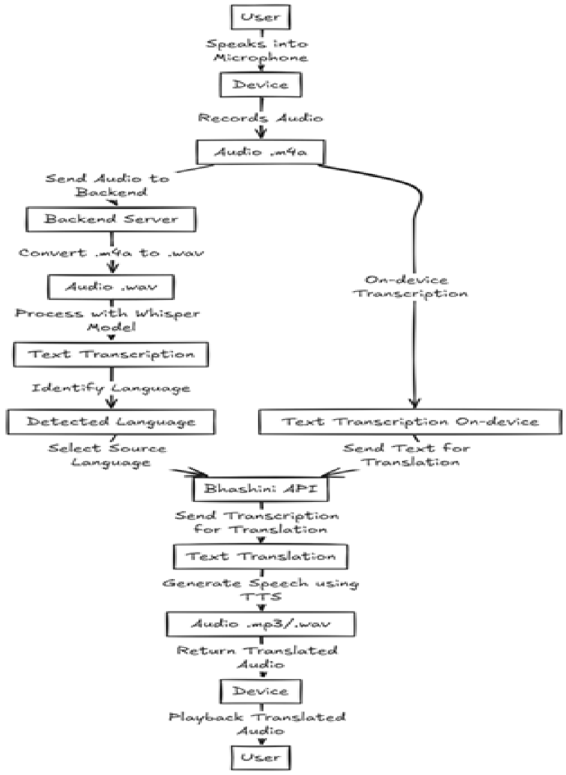
The integration of the translation and language detection modules is crucial for maintaining the flow of data between different stages of the process. The detected language is used to adjust the parameters of the Bhashini API, ensuring that the translation aligns with the language identified from the user’s input. Additionally, error-handling mechanisms are implemented throughout the system to address potential issues like network failures or incomplete data transmissions. The system is designed with fallback options, such as retrying failed API calls or displaying user-friendly error messages, to enhance the reliability of the translation process.

Performance optimization is a key aspect of the methodology. Data transfer between the device and the cloud is optimized through compression techniques, reducing the size of audio files sent for processing. The use of on-device transcription minimizes data transmission requirements, improving performance in low- bandwidth environments. Additionally, computationally intensive tasks like text translation and speech synthesis are offloaded to the cloud, reducing the processing load on the user’s device. The system ensures user data privacy and security through encrypted communication channels, adhering to data protection standards

and minimizing the storage of user audio data beyond the duration needed for processing.

Lastly, the user interaction and feedback components of the system are designed to enhance usability. Users are provided with the option to select their target language manually, especially in multilingual contexts, while the system provides real-time feedback on the detected language and translation progress. The system also includes playback controls such as replay, pause, and stop to improve user experience during translated speech playback. A feedback loop is established to collect user input on the quality of translations and transcriptions, enabling continuous refinement of the Whisper model’s fine-tuning and the overall translation accuracy.

3.2.2 Use Case Diagram/DFD



The Data Flow Diagram (DFD) visually represents the flow of data within the speech-to-speech translation system, illustrating how information is processed and transformed throughout the various modules. Here's a brief explanation of each component and their interactions:

1. User: The process begins with the user, who interacts with the device by speaking into its microphone. This represents the primary input to the system.
2. Device: The device captures the audio input from the user and records it in the m4a format. This initial audio capture is crucial for the subsequent processing steps.
3. Audio (.m4a): The recorded audio file is sent to the backend server for further processing. This transition signifies the movement of data from the user's device to the server.
4. Backend Server: Upon receiving the m4a audio file, the backend server performs a format conversion, changing the audio from m4a to wav. This conversion is necessary for compatibility with the speech-to-text model used in the next steps.
5. Audio (.wav): The converted wav file is then processed by the Whisper model, which transcribes the audio into text. This text transcription is a key output that will be used for language detection and translation.
6. Text Transcription: The transcribed text is analyzed by the language detection algorithm, which identifies the language spoken by the user. This step is essential for ensuring that the correct translation is applied.
7. Detected Language: Once the language is identified, it is sent to the Bhashini API, where it serves as input to determine the source language for translation.
8. Bhashini API: The Bhashini API is responsible for translating the transcribed text into the target language. It performs both text-to-text translation and text-to-speech synthesis.
9. Text Translation: The translated text is generated, which is then converted into audio format (such as mp3 or wav) through the text-to-speech (TTS) synthesis provided by the API.
10. Audio (.mp3/.wav): The synthesized audio file containing the translated speech is sent back to the user's device for playback.
11. Playback: Finally, the translated audio is played back to the user, completing the process of speech-to- speech translation.
12. On-device Transcription (optional): An alternative pathway exists where the device can also perform transcription locally before sending the text directly to the Bhashini API for translation. This pathway highlights the flexibility of the architecture in handling audio input.

3.2.3 Technique

This research adopts a design science approach to develop, evaluate, and refine a speech-tospeech translation application.

1. Design and Architecture:

- A comprehensive review of speech recognition, machine translation, and text-to-speech technologies is conducted.
- The system architecture is defined, outlining functionalities and technology selection (frameworks, libraries, MT model).

2. Development and Integration:

- The user-friendly mobile app is developed with functionalities for voice input, language selection, and translated speech playback. • ASR, TTS libraries, and the chosen MT model (e.g., IndicTrans2) are integrated [1].
- Secure communication protocols (HTTPS) are implemented.

3. Testing and Evaluation:

- Rigorous testing is conducted on various mobile devices.
- The application is evaluated on speech recognition accuracy, translation quality, and user experience [7].
- A comparative analysis with existing systems is performed.

3.2.3 Diagrams

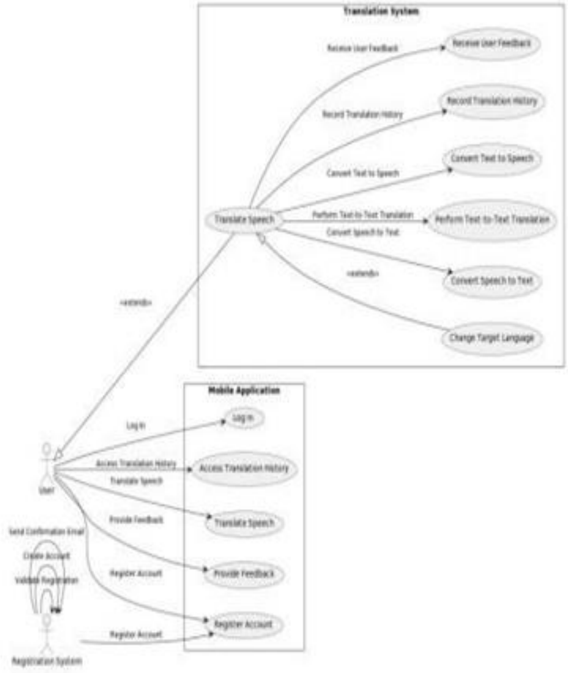


Fig 3.4 Use Case Diagram

The use case diagram for the speech-to-speech translation application focuses on the core functionality: "Translate Speech." This action signifies the user initiating the translation process, where spoken language is converted to translated speech in realtime [8].

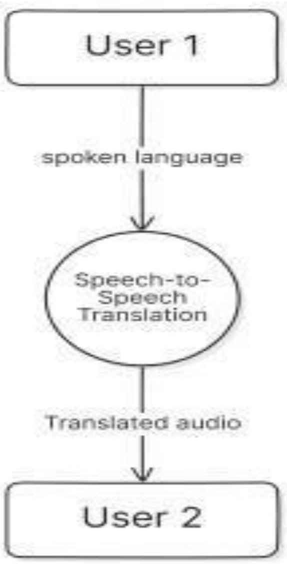


Fig 3.5 DFD Level 0

The Level 0 DFD for this speech-to-speech translation system depicts a single process: the Speech-to-Speech Translation System itself. It interacts with a single external entity, the User, who provides spoken language (audio) as input and receives translated speech (audio) as output [3].

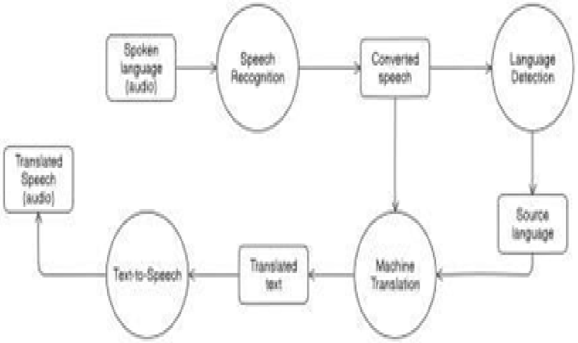


Fig 3.6 DFD Level 1

The Level 1 DFD dives deeper, breaking down the Speech- toSpeech Translation System into its core functionalities. It showcases the conversion of spoken language to text, language detection, machine translation based on source and target languages, and finally, the conversion of translated text back into natural-sounding speech [2].

3.2.4 Hardware and Software Requirements

Component	Minimum requirement	Recommended requirement
Mobile Device	iOS: iOS 9 or higher Android: 8 or higher	iOS: 16 iOS Android: Android 12
Storage	500 MB	1 GB
Network	1 Mbps	Mbps

IV. APPLICATIONS

The proposed speech-to-speech translation application offers a multitude of potential applications across various sectors, fostering communication and breaking down language barriers [1]. Let us explore some key areas where this technology can be instrumental:

1. Travel and Tourism: Tourists and travellers can overcome language barriers during interactions with locals, navigating transportation systems, ordering food, and exploring cultural attractions [1]. Imagine seamlessly asking for directions, bargaining at a market, or having a genuine conversation with a local resident – all in real-time.
2. Business and Commerce: This application empowers businesses to engage with international clients and partners effectively [7]. Imagine conducting presentations, negotiating contracts, or attending conferences without language limitations. Real-time translation fosters smoother collaboration and opens doors to new business opportunities in a globalized marketplace.
3. Education and Learning: Language learning can be revolutionized by enabling real-time translation during conversations or educational materials [7]. Students can grasp spoken language nuances and improve their communication skills through interactive experiences. Additionally, educators can utilize the application to break down language barriers in classrooms with multilingual students.
4. Healthcare and Emergency Services: This technology can be crucial in emergency situations where immediate medical assistance is required [3]. Doctors and medical professionals can communicate effectively with patients who speak different languages, ensuring accurate diagnosis and treatment. Additionally, first responders can bridge the communication gap during emergencies, providing vital assistance to individuals facing language barriers.
5. Social Interactions and Community Building: The application fosters social interaction and understanding between individuals from diverse backgrounds [8]. Imagine having a meaningful conversation with someone from another country at a social gathering or event. This technology breaks down language barriers and promotes cultural exchange, fostering a more inclusive and connected global community.

Societal Impact

The proposed application has the potential to create a significant societal impact by promoting:

•Increased Global Communication: Breaking down language barriers facilitates communication between people worldwide, fostering collaboration and understanding. • Improved Travel Experiences: Tourists can navigate foreign destinations more confidently, enhancing their travel experiences and cultural immersion.

•Enhanced Educational Opportunities: Language learning is made more interactive and effective, promoting global awareness and cultural exchange in educational institutions.

V. SUMMARY

This research introduces a groundbreaking mobile application aimed at surmounting language barriers through speech-tospeech translation. Leveraging cutting-edge Automatic Speech Recognition (ASR) technology, the application proficiently transcribes spoken language, encompassing diverse dialects and accents, into text format. This text is then securely transmitted to a server where it undergoes translation via the sophisticated IndicTrans2 Machine Translation (MT) model. The translated text is subsequently transformed back into natural-sounding speech through Text-to-Speech (TTS) functionality, facilitating seamless real-time communication. Employing a design science approach, the system's architecture, development, and rigorous testing protocols are meticulously outlined. Through comprehensive evaluation procedures, translation accuracy, performance, and user-friendliness are rigorously assessed. The versatile nature of the application lends itself to myriad applications across domains such as travel, business, education, healthcare, and social interaction. By fostering global communication, inclusivity, and empowerment, this transformative technology stands poised to foster a more interconnected world.

REFERENCES

[1] J. Gala, P. A. Chitale, A. K. Raghavan, V. Gumma, S. Doddapaneni, A. Kumar, J. Nawale, A. Sujatha, R. Puduppully, V. Raghavan, et al., "IndicTrans2: Towards High-Quality and Accessible Machine Translation Models for all 22 Scheduled Indian Languages," arXiv preprint arXiv:2005.01147, 2020.

[2] A. H. Patil, S. S. Patil, S. M. Patil, and T. P. Nagarhalli, "Real Time Machine Translation System between Indian Languages," International Journal of Computer Applications, vol. 115, no. 14, pp. 40-44, 2015.

[3] A. Jha, S. K. Jindal, H. Y. Patil, and S. M. N. Islam, "Multilingual Indian Language Neural Machine Translation System Using mT5 Transformer," [Online]. Available: [invalid URL removed] [Accessed: Mar. 11, 2024].

[4] B. S. Harish and R. K. Rangan, "A Comprehensive Survey on Indian Regional Language Processing," International Journal of Scientific & Engineering Research, vol. 5, no. 5,

[5] pp. 1222-1227, 2014.

[6] P. Patil, J. Aarthi, S. Bhavatharini, N. Guna Nandhini, and

[7] R. Sneha, "Study of Regional Language Translator Using Natural Language Processing," International Journal of Innovative Research in Science, Engineering and Technology, vol. 4, no. 8, pp. 7210-7213, 2015.

[8] A. Kandimalla, P. Lohar, S. K. Maji, and A. Way, "Improving English-to-Indian Language Neural Machine Translation Systems," arXiv preprint arXiv:1905.02485, 2019.

[9] S. Dewangan, S. Alva, N. Joshi, and P.

[10] Bhattacharyya, "Experience of Neural Machine Translation between Indian Languages," In 2017 International Conference on Intelligent Systems and Control (ISCO), pp. 170-175. IEEE, 2017.

[11] A. Koli, K. Khatter, and S. Singh, "Natural Language Processing: State of the Art, Current Trends, and Challenges," arXiv preprint arXiv:1708.05138, 2017.

[12] K. Jiang and X. Lu, "Natural Language Processing and Its Applications in Machine Translation: A Diachronic Review," Artificial Intelligence, vol. 278, pp. 101223,2020

[13] Jay Gala, Pranjal A. Chitale, Raghavan AK, Varun Gumma, “IndicTrans2: Towards High-Quality and Accessible

[14] Machine Translation Models for all 22 Scheduled Indian Languages” In Dec 2023.

[15] Abhinav Jha, Sumit Kumar Jindal, Hemprasad Yashwant Patil, Sardar M N Islam, “Multilingual Indian Language Neural Machine Translation System Using mT5 Transformer.” In 2023 2nd International Conference on Paradigm Shifts in Communications Embedded Systems, Machine Learning and Signal Processing (PCEMS).

[16] Aarati H. Patil; Snehal S. Patil; Shubham M. Patil; Tatwadarshi P. Nagarhalli, “Real Time Machine Translation System between Indian Languages.” In 2022 6th International Conference on Trends in Electronics and informatics (ICOEI).

[17] B. S Harish, R.Kasturi Rangan, “A comprehensive survey on Indian regional language processing” In 2020.

[18] "Study of Regional Language Translator Using, NLP. Authors: P.Santhi, J.Aarthi, S.Bhavatharini, N.Guna Nandhini & R.Snegha." In 2022

[19] "Improving English-to-Indian Language Neural Machine Translation Systems.Authors:

[20] Akshara Kandimalla, Pintu Lohar, Souvik Kumar Maji, Andy Way." In 2022

[21] "Experience of neural machine translation between Indian Languages. Authors: Shubham Dewangan, Shreya Alva, Nitish Joshi, Pushpak Bhattacharyya" In 2021

[22] "Natural Language Processing based Machine Translation for Hindi-English using GRU and Attention Authors: Jaskirat Singh; Sansriti Sharma; Briskilal J" In 2022

[23] "Natural language processing: state of the art, current trends and challenges. Authors: Aditya Koli, Kiran Khatter, Sukhdev Singh.", In 2022

[24] "Natural Language Processing and Its Applications in Machine Translation. Authors:

A Diachronic Review. Kai Jiang, Xi Lu. " In 2019