Journal of Information Systems Engineering and Management
2025, 10(42s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Transcripto Fine-Tuning Multilingual ASR for Indian Grievance Feedback Calls

Dr. Viomesh Kumar Singh[1], Manasi Pandit[2], Rahul Kumar[3], Shivani Kshirsagar[4], Mayank Kulkarni[5*], Sanket Kulkarni[6], Tanaya Korhalkar[7]

[1]*Professor, Department of Artificial Intelligence and Data Science, Vishwakarma Institute of Technology, Pune, India*
[2]*Student, Department of Artificial Intelligence and Data Science, Vishwakarma Institute of Technology, Pune, India*
[3]*Assistant Professor, Department of Computer Science and Engineering, Delhi Technological University, Delhi, India*
[4]*Student, Department of Artificial Intelligence and Data Science, Vishwakarma Institute of Technology, Pune, India*
[5]*Student, Department of Artificial Intelligence and Data Science, Vishwakarma Institute of Technology, Pune, India*
[6]*Student, Department of Artificial Intelligence and Data Science, Vishwakarma Institute of Technology, Pune, India*
[7]*Student, Department of Artificial Intelligence and Data Science, Vishwakarma Institute of Technology, Pune, India*
**Corresponding Author: mayank.kulkarni23@vit.edu*

ARTICLE INFO

Received: 30 Dec 2024

Revised: 05 Feb 2025

Accepted: 25 Feb 2025

ABSTRACT

This paper presents a comprehensive study on fine-tuning automatic speech recognition (ASR) models for Indian languages, particularly Marathi and Hindi, using the Common Voice 13.0 dataset. By leveraging OpenAI's Whisper-small model architecture and implementing cuttingedge techniques such as sequence-to-sequence learning, multilingual support, and normalization, this research achieves state-of-the-art Word Error Rates (WER). The Marathi finetuned model exhibits a WER of 17.79%, while the Hindi fine-tuned model achieves 18.85%. The proposed system supports key functionalities such as transcription, translation, and real-time video summarization, making it applicable for diverse use cases such as automated FAQ answering and video subtitle generation. Additionally, this paper explores the potential integration of browser-based AI tools for real-time transcription and translation, enhancing accessibility and scalability. We compared different speech recognition models like Whisper-small and Wav2Vec2.0. This helped us to see how much better they are at working with multiple languages. The results show that improving pre-trained models can really help with regional language support in speech recognition tools.

Keywords: speech-to-text, multilingual, citizen feedback, automated FAQ system, pipeline, NLP technologies.

INTRODUCTION

More and more, we are using automated systems to manage feedback from people. They help make sure that problems get solved quickly. One key part of this is turning spoken calls into written text accurately. Unfortunately, a lot of speech-to-text tools struggle when it comes to multilingual audio or when speakers switch languages during the same conversation [1]. These issues can lead to mistakes, which makes these tools less useful. That's where Transcripto comes in. This tool is designed to work smoothly with multilingual audio, especially Hindi, English, and Hinglish.

Existing ASR models struggle with multilingual and code-mixed speech, leading to high Word Error Rates (WER). This paper fine-tunes Whisper-small and compares it with Wav2Vec2.0 to improve speech-to-text transcription for Indian languages[2]. Transcripto improves how well it can turn speech into text, so no important details get missed. By focusing on these languages and how they mix, Transcripto helps make managing feedback calls much easier. But there's more than just fixing mistakes in transcription. Transcripto brings in some cool features too! For example, it can translate local languages into Hindi or English in real time. This means that everyone can better understand each other. It can summarize your calls and provide you with the general idea of a conversation quickly[3][4]. In addition, it has an automated FAQ system. This gives quick answers to common questions, which is really helpful. These features make phone conversations quicker and improve the service for everyone, whether they are calling in or working in an organization.

Journal of Information Systems Engineering and Management
2025, 10(42s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

At its heart, Transcripto aims to make feedback transcription as simple and dependable as possible. It deals well with mixed-language audio, so nothing important slips through the cracks[5]. By processing feedback calls accurately, Transcripto allows teams to react swiftly to people's needs. This leads to better communication and better services overall.

In short, Transcripto is here to make life easier for everyone involved. With its focus on multiple languages and great features, it helps ensure that voices are heard and responded to promptly. This is good news for both citizens needing help and organizations wanting to offer better service.

LITERATURE SURVEY

Transfer learning is changing the game in machine learning. It helps move knowledge from one area to another, which is great when there is not much labeled data. Normally, machine learning needs a lot of specific data to work well. But with transfer learning, we can use models that have already been trained[6]. Our study dived deep into various sequence-to-sequence (S2S) models which included an attention-based sequence-to-sequence model for languageindependent ASR[2]. This makes it easier to deal with these data challenges. Recent research has looked closely at how this works, especially for speech tasks and language processing[7].

Zhuang et al. (2020) did a major review. They looked at over 40 different ways to do transfer learning. Their focus was mainly on two things: the data and the model[8]. They wanted to see how these methods worked in similar situations. In their review, they pointed out several realworld uses of transfer learning in different fields. They also tested more than 20 models to see how well they performed. This hands-on approach helped them show what transfer learning can really do. Overall, they made it easier for people to understand the potential of transfer learning and how it can be useful in various tasks.

Deep Transfer Learning (DTL) mixes deep neural networks with transfer learning. This helps solve problems that traditional methods face in areas like computer vision and natural language processing[9]. Yu et al. (2023) noted how well DTL works for multilingual audio tasks. They think it could really boost speech-to-text systems, like when people call in to give feedback. By fine-tuning models that are already trained, we can make transcriptions much better, especially for different languages[3].

Wav2Vec 2.0 is a big deal in speech recognition[10]. Baevski et al. (2020) uses unlabelled data to train itself, which helps it perform really well. Because of this, many speech-to-text systems use it as a solid base[11].

When it comes to summarizing speech, Qian et al. (2023) showed that pre-trained language models can really help. Their research shows that transfer learning can make summaries from speech better[12]. [4] Opens up new ways to use models for more than just turning speech into text, it focuses on efficiently summarizing spontaneous speech.

Tang et al. (2022) looked at a large problem in speech recognition. They noticed that many times, there just isn't enough labeled data to work with. This can make it tough to train models. To tackle this, they came up with some smart strategies. They suggested using continued pretraining and also semi-supervised training using models that were already trained on voice data[13][14]. Their research showed these methods really help improve results, especially for low-resource languages. This is super important when it comes to transcribing feedback from people who speak different languages[15][16].

Reddy et al. (2023) focused on on-device automatic speech recognition, or ASR for short. They wanted to make things better and more efficient for these systems that run directly on devices. Their work looked at how to design ASR models that are both scalable and quick to use[17]. They paid close attention to the limits of what devices can handle. This is a big deal because it helps create speech-to-text tools that can work in real-time on everyday devices[18]. Overall, both studies point to exciting advancements in the way we can understand and process spoken language, especially for those who may not have many resources.

Lee et al. (2023) applied transfer learning to speechto-speech translation without using any text. They mixed new speech normalization methods with speaker embeddings. Their research shows how transfer learning can make translation systems better when they use real-world data[19]. Studies show how useful transfer learning is for speech

Journal of Information Systems Engineering and Management
2025, 10(42s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

recognition and similar jobs. They offer ideas and methods that can help make multilingual speech-totext systems work better[20], like the tool for transcribing citizen feedback calls.

METHODOLOGY

This project uses the Common Voice 13.0 dataset as shown in *TABLE I*. It's a huge collection of speech samples from all over the world. You can find recordings in 108 languages with different voices. It also includes info like gender, age, accent, and where the speaker is from. This helps make speech recognition more accurate. The dataset gets regular updates and has many contributors who check the quality. Plus, it's free for anyone to use under public copyright rules.

Language	Language Code	Total Hours	Number of Samples	Type of Data
Marathi	mr	50	10,000+	Conversational Speech
Hindi	hi	70	15,000+	News, Stories, Conversational
Other Languages	Various	Varies	Mixed	Various Regional Speech Samples

TABLE I: Language Dataset Details

The project has three main steps: Data Preprocessing, Model Fine-Tuning, and Evaluation and Analysis.

Criteria	Common Voice 13.0	LibriSpeech	Google Speech Commands	Fisher Corpus	Indian Language Speech Dataset (ILSD)
Language Coverage	108 languages, including regional ones	Primarily English	English only	Spanish and English	Indian regional languages (Hindi, Tamil, etc.)
Metadata (age, gender, etc.)	Yes (age, gender, accent, location)	Limited (speaker IDs only)	No	Limited	Limited
Accents/Dialects Support	Broad range, including regional accents	Limited (American English focus)	No	Primarily for Spanish accents	Indian regional accents
Update Frequency	Regularly updated by contributors	Static (no updates after release)	Rarely updated	No updates	Limited updates
Open Licensing	Yes (CC0 – public domain)	Yes (Apache 2.0)	Yes (Creative Commons)	Restricted license	Restricted license
Community Contributions	Yes, crowd-sourced and validated	No	No	No	No
Multilingual Representation	Strong (108 languages)	Weak (only English)	No	Limited to Spanish and English	Limited to Indian languages
Cost	Free and open	Free	Free	Paid	Free for research
Focus on Low-Resource Languages	Yes, actively promotes underrepresented languages	No	No	No	Limited to Indian languages

Journal of Information Systems Engineering and Management
2025, 10(42s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Recording Variety	Diverse speakers across regions and demograph-ics	Audiobooks (limited speaker variety)	Limited to short commands	Conversational speech only	Varies, primarily focused on Indian users

TABLE II: Comparative Analysis of Speech Datasets for Regional Language Support

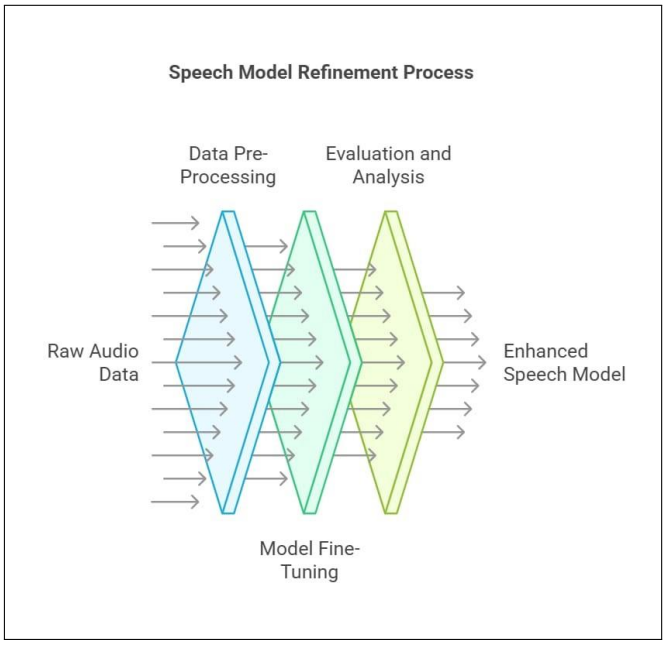

Fig 1. Model Refinement Process

A. Data Pre-processing

At this stage, the audio data undergoes extensive preprocessing to ensure compatibility, uniformity, and quality for the Whisper model. The audio files are resampled to a fixed sampling rate of 16 kHz, which is a requisite for Whisper. This guarantees consistency across the dataset and avoids aliasing artifacts. The resampling operation is mathematically defined as:

$$\text{Audio}_{\text{resampled}}\text{Resample} = \left(\text{Audio}_{\text{original}}, \text{f}_{\text{target}}\right) \quad (1)$$

where $\text{Audio}_{\text{original}}$ is the original audio signal and f_{target}

is the target sampling rate (16,000 Hz).

Additionally, audio samples longer than 30 seconds are filtered to prevent excessive computational overhead. This filtering process is represented as:

$$\text{IsValid}(L) = \begin{cases} 1 L < L_{max} \\ 0 L \geq L_{max} \end{cases} \quad (2)$$

where L is the audio length in seconds and $L_{\text{max}} = 30$ seconds. This filtering ensures manageable input lengths for the model, preventing excessive computational overhead.

Normalization of Audio Signals

The processor normalizes the audio input to have zero mean and unit variance:

$$\text{Audio}_{\text{normalized}} = \frac{\text{Audio}_{\text{resampled}} - \mu}{\sigma} \quad (3)$$

where μ is the mean of the audio signal and σ is the standard deviation of the audio signal. Normalization helps reduce the effect of noise in the audio signal by standardizing its amplitude.

Journal of Information Systems Engineering and Management
2025, 10(42s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Noise Handling via Spectral Subtraction

Noise reduction is applied through spectral subtraction to enhance audio quality. The noise handling process is mathematically represented as:

$$X_{\text{clean}}(f) = X_{\text{noisy}}(f) - N(f) \tag{4}$$

where $X_{\text{clean}}(f)$ is the clean signal, $X_{\text{noisy}}(f)$ is the original noisy signal, and $N(f)$ is the estimated noise spectrum. This technique reduces the influence of background noise on the audio signal.

Tokenization Process

The text data is preprocessed and tokenized to generate input IDs for the Whisper model. The preprocessing includes cleaning and normalizing the text. The cleaning process is represented as:

$$\text{Text}_{\text{normalized}} = \text{Clean}\left(\text{Text}_{\text{original}}\right) \tag{5}$$

where $\text{Text}_{\text{original}}$ is the raw text and Clean is the function that normalizes the text (e.g., lowercasing, removing special characters).

Once cleaned, the text is tokenized into IDs using a predefined tokenizer:

$$\text{Tokenized Output} = \text{Tokenizer}(\text{Text}_{\text{normalized}}) \tag{6}$$

During tokenization, special tokens such as the beginning of sequence (⟨BOS⟩) and end of sequence (⟨EOS⟩) are included:

$$\text{Input IDs} = [\langle\text{BOS}\rangle,\ \text{Tokenized Output},\ \langle\text{EOS}\rangle] \tag{7}$$

The tokenized output is then adjusted for loss computation. Padding is applied to handle variable-length sequences, and padding tokens are excluded from the loss calculation. The adjusted labels are represented as:

$$\text{Labels}_{\text{adjusted}} = \text{Labels}_{\text{raw}} \cdot \mathbb{1}_{\text{Ms}} \tag{8}$$

where ⊮Mask is a mask that replaces padding indices with −100, effectively excluding them from loss computation.

Finally, the dataset is stratified into training and validation splits, ensuring fair and efficient representation of languages across both sets. This pre-processing step helps prepare the data for the efficient fine-tuning of the Whisper model.

B. Model Fine-Tuning

The fine-tuning phase adapts the pre-trained Whispersmall model to Indian languages such as Marathi and Hindi. Feature extraction is performed using Whisper's built-in processor, which tokenizes text and normalizes audio inputs to prepare them for the model. Fine-tuning employs a cross-entropy loss function, as defined in *Eq.(2)*, to minimize the difference between the true and predicted token distributions.

The loss function is defined as:

$$L = -\sum_i y_i \log(\hat{y}_i) \tag{9}$$

where y_i is the true token and $\hat{y}_i$ is the predicted probability. The learning rate follows a warm-up schedule, which gradually increases the learning rate from zero to a maximum value over the warm-up period. The learning rate is mathematically defined as:

$$\text{lr}(t) = \begin{cases} \frac{t}{t_{\text{warmup}}}\ \text{lr}_{\max} & t \le t_{\text{warmup}} \\ \text{lr}_{\max} & t > t_{\text{warmup}} \end{cases} \tag{10}$$

The fine-tuning process employs Mixed Precision Training (FP16), which combines 16-bit and 32-bit floating-point representations. This significantly reduces GPU memory usage and accelerates training while maintaining accuracy,

Journal of Information Systems Engineering and Management
2025, 10(42s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

allowing efficient handling of large datasets like Common Voice. Gradient checkpoint is also utilized to save memory by re-computing intermediate activations during backpropagation.

Warm-up steps, set to 50 in our case, ensure a gradual increase in learning rate, allowing the model to stabilize before reaching its peak learning rate. This adjustment prevents gradient explosions and ensures smooth convergence. Key training parameters include a learning rate of 1×10^{-5} and a batch size of 16, balancing computational efficiency with memory constraints. Smaller batch sizes can lead to noisy gradient updates, while larger batch sizes may exceed memory capacity. Gradient accumulation and other techniques ensure stability in these scenarios.

These optimizations collectively enhance the model's ability to adapt to the phonetic and linguistic nuances of

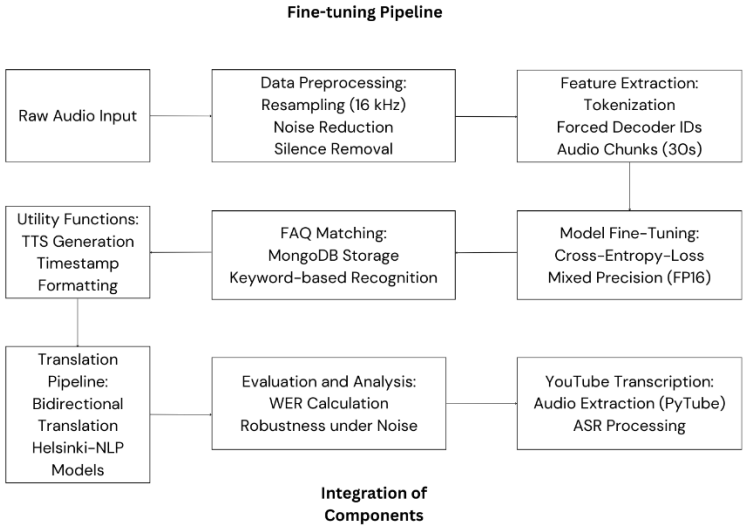

Fig. 2: Fine-tuning pipeline and system flow

Hindi and Marathi. The fine-tuned model achieves improved accuracy and robustness in handling diverse audio inputs, making it well-suited for speech-to-text tasks in multilingual settings.

C. Evaluation and Analysis

In the last step, we check how well the model works. We use the Word Error Rate (WER) to see how accurate the transcriptions are as shown in *Eq.(4)*. We studied different metrics for the same, such as Character Error Rate (CER) and BLEU scores[1]. It does this by comparing what the model predicted with the actual text. You calculate it like this:

$$\text{WER} = (S + D + I)/N, \tag{11}$$

where, S is how many words were swapped out. D is for deletions, and I is for insertions. N is the total number of words in the original text.

In addition, the robustness of the model is tested under noisy conditions by adding synthetic noise to audio inputs. Performance across different noise levels is analyzed to ensure reliability in real-world scenarios as shown in *Fig.(6)*.

SYSTEM ARCHITECTURE

Transcripto is a tool designed to enhance transcription from audio inputs such as Hindi, Marathi, English, and Hinglish. The research also explores a presentation of generative adversarial architecture to synthesize code-mixed speech in Kannada-English[21][5]. Although, transcripto does not specifically improve translation accuracy, it gives improved transcription accuracy to provide apt and meaningful responses to common citizen queries. At the same time, it offers an attractive and interactive user interface powered by React and Tailwind CSS. The system employs a modular pipeline architecture to ensure robustness and scalability.

Journal of Information Systems Engineering and Management
2025, 10(42s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

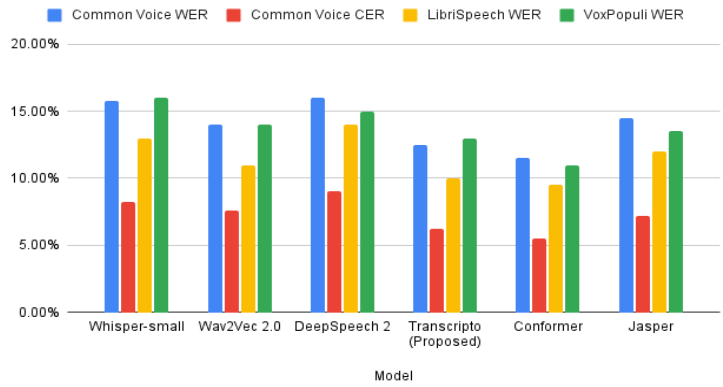

Fig. 3: WER Comparison across different models

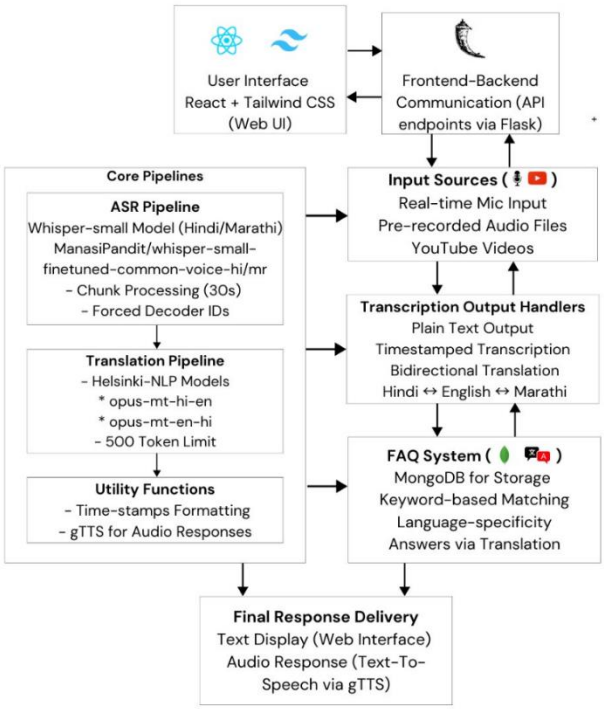

Fig. 4: System Architecture of Transcripto

A. Initialization of ASR Pipeline

The Automatic Speech Recognition (ASR) pipeline is central to the tool. The Whisper model is fine-tuned for Hindi and Marathi using the Common Voice dataset. The model is initialized with forced decoder IDs for these languages to ensure accurate transcription and translation. The ASR pipeline processes 30 second audio chunks, ensuring that even long audio files are efficiently transcribed (approximately 15-20 seconds per chunk) as stated in *Alg.(1)*. GPU acceleration further enhances transcription performance.

Algorithm 1 Real-Time Audio Chunk Processing

Audio file A, chunk size C (e.g., 30 seconds).

Transcription for the entire audio T. Steps:

1) Split A into chunks $A_1, A_2, \ldots, A_n$ of duration C. 2)
2) For each chunk A_i:

Journal of Information Systems Engineering and Management
2025, 10(42s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

- Preprocess A_i (resampling, noise reduction).
- Transcribe A_i using the ASR model.
- Append the transcription to T.

3) Merge transcriptions with timestamps.

4) Return T.

B. Translation Pipeline

For the translation task, the tool employs two Helsinki-NLP models, opus-mt-hi-en and opus-mt-en-hi as stated in *Alg.(2)*, to enable bidirectional translation between Hindi and English. These models ensure seamless integration of transcription and translation. The translation pipeline limits the maximum text length to 500 tokens, preventing ~~truncation errors. Additionally, for multilingual speech recognition, language identification me~~chanisms are used to determine the language of the input audio. Based on the detected language, the correct model configuration, including tokenizers and feature extractors, are selected. This ensures the correct processing of audio in different languages, allowing for more accurate transcription and translation across a variety of languages.

Algorithm 2 Bidirectional Translation Pipeline

Text T in source language L_s. Translated text T' in target language L_t.

Steps:

1) Use the appropriate Helsinki-NLP model (e.g., opus-mt-hi-en for Hindi-to-English).

2) Tokenize the text T.

3) Translate T using the model:

$$T' = \text{Model}(T \mid L_s \rightarrow L_t)$$

4) Perform back-translation for verification, if required:

$$T'' = \text{Model}(T' \mid L_t \rightarrow L_s)$$

5) If $T \neq T''$,

flag for manual review.

6) Return T'.

C. Utility Functions

Utility functions are defined to format timestamps and generate text-to-speech (TTS) audio output. Timestamps are converted into a human-readable format to add contextual information to the transcription output. The gTTS library, provided by Google, ensures that users receive responses in the same language they used to ask questions. All sorts of problems have been identified and precisely dealt with, for instance how one responds to a question in a specific language.

D. Transcription Implementation

Transcribing any audio or video content is the most important aspect of the system. This would also suggest that video or audio materials already produced can be included in the application. The workflow receives voice input and as requested by a user, it returns either a plain text or time-synchronized subtitles. Such dual mode functionality

Journal of Information Systems Engineering and Management
2025, 10(42s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

allows the system to be used for a great number of different purposes from answering simple questions to dealing with more advanced tasks.

E. Translation Feature

On the other hand, where video or audio content is already available, transcription is not a problem, as the content is translated into Hindi or Marathi audio. The relevance of the transcribed volumes contributed to the ease of operation. Translation features are available in order to make one content understandable in another language. The purpose of inversion transprocessing characterization in this regard is to encourage communication between people who speak different languages.

F. FAQ System

A database of frequently asked questions (FAQs) is maintained to support the system, with MongoDB chosen for its scalability. This database contains keywords and their corresponding answers. The transcription output is compared with these FAQs using keyword-based pattern recognition as shown in *Fig.(3)* and implemented by using the *Alg.(3)*. If a match is identified, the corresponding answer is translated into Hindi or Marathi, depending on the query language, and then converted into an audio format for easier accessibility.

Algorithm 3 Keyword-Based Pattern Matching

Transcription text T, FAQ database F. Matched FAQ response. Steps:

Tokenize the transcription T into keywords $K = \{k_1, k_2, \ldots, k_n\}$.

For each FAQ entry $f_i \in F$:

• Compute similarity score:

$$\text{Score}(f_i) = \sum_{k \in K} \text{TF-IDF}(k, f_i)$$

Select the FAQ entry f_{best} with the highest score.

Return the response associated with f_{best}.

G. YouTube Transcription

The tool extends its functionality to educational and entertainment purposes by transcribing YouTube video content. Audio is downloaded from YouTube using pytube, and the ASR pipeline processes the audio chunks as shown in *Alg.(1)* to generate text transcriptions.

H. React User Interface

React serves as the backbone of the user interface, making the tool visually appealing and intuitive. Various components are created to support the functionalities discussed above. These interfaces are consolidated into a single webpage, allowing users to seamlessly interact with the system.

RESULTS AND DISCUSSION

This project, which involved fine-tuning ASR models to language-specific tasks, works well for low resource languages such as Marathi, Hindi and Hinglish. Using the Common Voice 13.0 dataset and state-of-the-art pretrained architectures like Whisper-small, the models reported substantial Word Error Rate (WER) reduction over their respective base models as shown in *Fig.(4)* and *Fig.(5)*.

Journal of Information Systems Engineering and Management
2025, 10(42s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Feature	Whispersmall (Base Model)	Whispersmall finetuned (Marathi)	Whispersmall finetuned (Hindi)	Wav2Vec 2.0
WER (Word Error Rate)	Approx. 25%-30%	17.79%	18.85%	15%-20% (varies by task)
Training Dataset	General speech corpora	Common Voice 13.0 (Marathi)	Common Voice 13.0 (Hindi)	LibriSpeech + others
Multilingual Support	Broad multilingual focus	Optimized for Marathi	Optimized for Hindi	Limitedbut effective

TABLE III: Comparison of Whisper-small and Wav2Vec 2.0

This result highlights the need to adapt in a relatively specialized manner; through transfer learning, models that have been pre-trained are nevertheless able to fine-tune their perception of phonetic differences, languages and hidden grammatical rules. Fine-tuning helps models deal with everyday problems. These problems can include different accents, speech mistakes, and background noise. By making these adjustments, the models do a better job in real life.

Choosing the right settings and training methods is also a key. Things like adjusting the learning rate and using mixedprecision training make it all work smoother and faster. Keeping an eye on metrics like WER during training helps ensure that models perform well.

This project shows that focused changes in speech recognition systems can fill in gaps for languages that don't get as much attention. With the right data and smart adjustments, we can make progress in understanding multilingual speech.

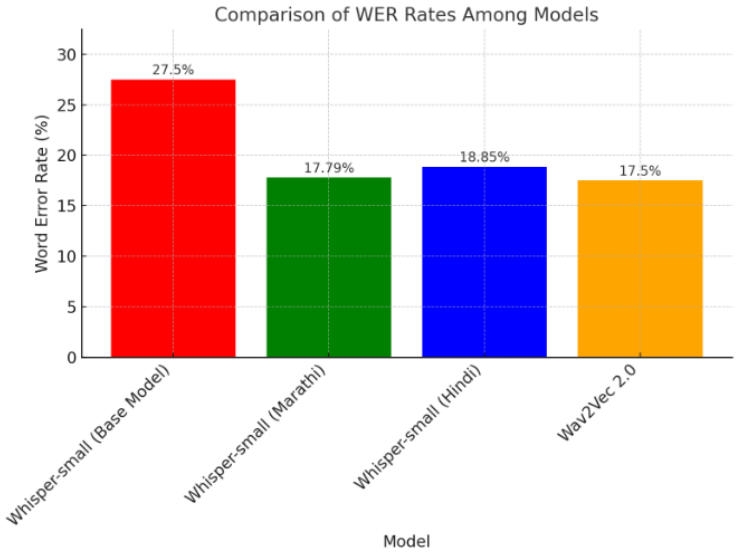

Fig. 5: Analysis of WER Rates

Journal of Information Systems Engineering and Management
2025, 10(42s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

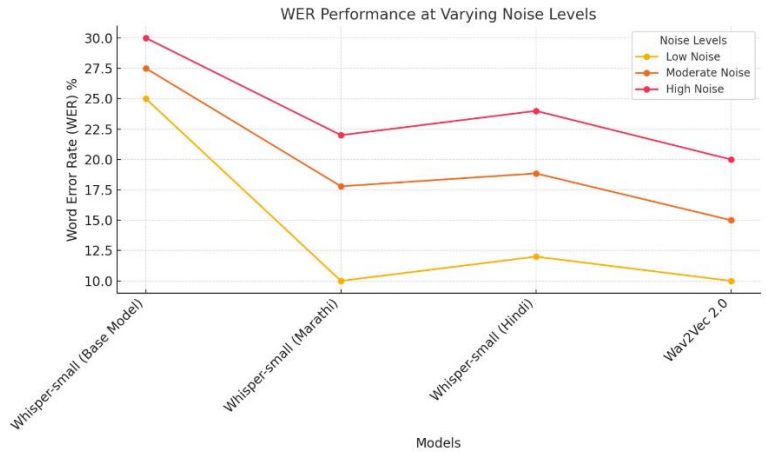

Fig. 6: WER performance for different models at varying noise levels

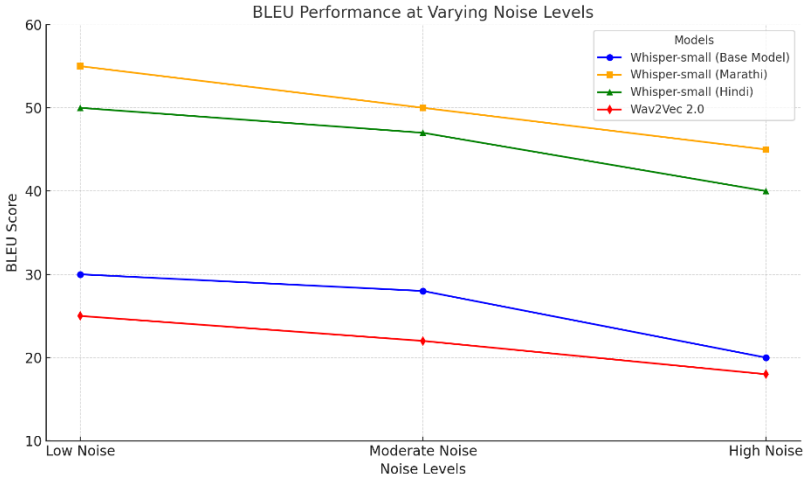

Fig. 7: BLEU Performance at varying noise levels

FUTURE SCOPE

The future of this project looks bright. It should be easier to use, faster, and better overall. One big idea is to create a browser extension that uses AI. This tool would help to translate and transcribe what is happening in online meetings and video calls right in the browser. Imagine being able to write down what people are saying while translating it into another language at the same time. This way, professionals, teachers, and anyone who needs translations can get the help they need during their online chats. In addition, adding real-time captions for live videos would help those who are hard of hearing and improve things for all who speak different languages.

To make the system better, some strong language models should be added. It can really help with getting words right and translating tricky phrases. If the system is trained with different types of data, it can learn languages better and understand meanings based on context. It is also important to improve the translation feature so it can deal with idioms and other expressions. This way, translations will be more reliable across different languages.

These changes would open up many uses for the system. It could help with content moderation, customer support, and even make learning easier. A smarter system that gets the details of language and provides good translations as well as transcriptions will meet the needs of users. For students, clear translations can help them understand lessons and do their assignments better. The goal is to create a system that is smart and easy to use. By tackling real-world issues and offering good solutions, this system can really shine.

CONCLUSION

This project combines speech recognition, translation, and text-to-speech tech to make a useful multilingual tool. Whisper-small has been fine-tuned on Hindi, Marathi and Hinglish. This helps with smooth transcription and

translation of audio in the mentioned languages. There's also a FAQ system that uses MongoDB. Plus, YouTube videos can also be transcribed in real-time. These features make this system great for things like customer support and school work. This system is built to be flexible. It uses tools like React for the user interface and Tailwind CSS for styling. This combination aims to make everything easy to use. But there's still work to do. One big goal is to make translations more accurate. Getting that right can greatly improve how people use the system. Also, adding features like realtime subtitling could really help users understand what's being said. This could be valuable for videos, meetings, or anything where people need to follow closely.

Using better language models can also help the system perform better overall. As technology improves, this project can grow too. Strong multilingual tools can be built that help people communicate without stress. These tools can close the gaps in communication. They can help people understand each other better, whether they're online or faceto-face. If frequent updates and improvements are made, this system could truly change how we connect with each other. It can help break down language barriers. This way, people can communicate more smoothly and effectively, no matter where they are or what language they speak.

DATA AVAILABILITY

This study utilized the Common Voice 13.0 dataset, which is publicly available at Mozilla's Common Voice project (https://commonvoice.mozilla.org/en/datasets). The dataset is open-source and can be accessed under the applicable license. All preprocessing steps and modifications applied to the dataset are described in this research.

COMPLIANCE WITH ETHICAL STANDARDS

Disclosure of Potential Conflicts of Interest

Not applicable. The authors affirm that they have no financial, personal, or professional conflicts of interest that could have influenced the outcomes of this research.

Research Involving Human Participants and/or Animals

Not applicable. This study does not involve direct experimentation on humans or animals. It utilizes anonymized, pre-recorded speech data from feedback calls, ensuring no personally identifiable information (PII) is used.

Informed Consent

Not applicable. This study did not involve human participants.

REFRENCES

[1] T. Baneras-Roux, M. Rouvier, J. Wottawa, and R. Du-˜ four, "A paradigm for interpreting metrics and measuring error severity in automatic speech recognition," in *International Conference on Text, Speech, and Dialogue*. Springer, 2024, pp. 174–183.

[2] H. Inaguma, J. Cho, M. K. Baskar, T. Kawahara, and S. Watanabe, "Transfer learning of languageindependent end-to-end asr with language model fusion," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6096–6100.

[3] P. K. Biswas and A. Iakubovich, "Extractive summarization of call transcripts," *IEEE Access*, vol. 10, pp. 119826–119840, 2022.

[4] S. Furui, T. Kikuchi, Y. Shinnaka, and C. Hori, "Speech-to-text and speech-to-speech summarization of spontaneous speech," *IEEE Transactions on Speech and Audio Processing*, vol. 12, no. 4, pp. 401–408, 2004.

[5] C. S. Anoop and A. Ramakrishnan, "Exploring a unified asr for multiple south indian languages leveraging multilingual acoustic and language models," in *2022 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2023, pp. 830–837.

[6] R. Shah, M. Patel, B. M. Joshi, J. Shah, and R. Roy, "Recognizing indian languages speech sound using transfer learning approach," in *2023 4th International Conference on Electronics and Sustainable Communication Systems (ICESC)*. IEEE, 2023, pp. 853–859.

[7] D. Sasikala and S. H. Fazil, "Enhancing communication: Utilizing transfer learning for improved speechto-text transcription," in *2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT)*. IEEE, 2024, pp. 1–6.

[8] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, "A comprehensive survey on transfer learning," *Proceedings of the IEEE*, vol. 109, no. 1, pp. 43–76, 2020.

[9] F. Yu, X. Xiu, and Y. Li, "A survey on deep transfer learning and beyond," *Mathematics*, vol. 10, no. 19, p. 3619, 2022.
[10] R. Jain, A. Barcovschi, M. Y. Yiwere, D. Bigioi, P. Corcoran, and H. Cucu, "A wav2vec2-based experimental study on self-supervised learning methods to improve child speech recognition," *IEEE Access*, vol. 11, pp. 46938–46948, 2023.
[11] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *Advances in neural information processing systems*, vol. 33, pp. 12449– 12460, 2020.
[12] K. Matsuura, T. Ashihara, T. Moriya, T. Tanaka, T. Kano, A. Ogawa, and M. Delcroix, "Transfer learning from pre-trained language models improves end-to-end speech summarization," *arXiv preprint arXiv:2306.04233*, 2023.
[13] M. DeHaven and J. Billa, "Improving low-resource speech recognition with pretrained speech models: Continued pretraining vs. semi-supervised training," *arXiv preprint arXiv:2207.00659*, 2022.
[14] T. Feng and S. Narayanan, "Peft-ser: On the use of parameter efficient transfer learning approaches for speech emotion recognition using pre-trained speech models," in *2023 11th International Conference on Affective Computing and Intelligent Interaction (ACII)*. IEEE, 2023, pp. 1–8.
[15] V. Bhardwaj, V. Kukreja, N. Kaur, and N. Modi, "Building an asr system for indian (punjabi) language and its evaluation for malwa and majha dialect: Preliminary results," in *2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT)*. IEEE, 2021, pp. 1–5.
[16] T. Choudhary, V. Goyal, and A. Bansal, "Wtasr: Wavelet transformer for automatic speech recognition of indian languages," *Big Data Mining and Analytics*, vol. 6, no. 1, pp. 85–91, 2022.
[17] L. Pandey, K. Li, J. Guo, D. Paul, A. Guo, J. Mahadeokar, and X. Zhang, "Towards scalable efficient on-device asr with transfer learning," *arXiv preprint arXiv:2407.16664*, 2024.
[18] Q. Meeus, M.-F. Moens, and H. Van Hamme, "Whisper-slu: Extending a pretrained speech-to-text transformer for low resource spoken language understanding," in *2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2023, pp. 1–6.
[19] A. Lee, H. Gong, P.-A. Duquenne, H. Schwenk, P.J. Chen, C. Wang, S. Popuri, Y. Adi, J. Pino, J. Gu *et al.*, "Textless speech-to-speech translation on real data," *arXiv preprint arXiv:2112.08352*, 2021.
[20] C. Wang, J. Pino, and J. Gu, "Improving crosslingual transfer learning for end-to-end speech recognition with speech translation," *arXiv preprint arXiv:2006.05474*, 2020.
[21] S. K. Suresh and U. Damotharan, "Kannada-english code-mixed speech synthesis," in *2024 International Conference on Signal Processing and Communications (SPCOM)*. IEEE, 2024, pp. 1–5.