Automatic Speech Recognition in Indic Languages with enhanced Spell Checker using Masked Language Modelling

RAJALAKSHMI R, Vellore Institute of Technology Chennai, India
SUJITH M, Vellore Institute of Technology Chennai, India

Automatic speech recognition involves the detection of speech from audio and converting it into text. This research focuses on advancing the capabilities of Automatic Speech Recognition (ASR) for the low resource Indic Languages such as Tamil, Malayalam, Marathi and Kannada. We employed IndicSpellFixASR (ISFASR), a model finetuned from the whisper model from OpenAI for the task of speech recognition and integrated with our own proposed spell corrector. We evaluate the efficacy of our fine-tuned model through rigorous testing and performance metrics such as word and character error rates to check its suitability for real-world applications such as transcription services, voice assistants, and other voice-driven technologies. The ASR model's prediction errors are further minimized by incorporating the XLM-Roberta-Large model, leveraging masked language modelling, and utilising the Difflib library for word matching. The usage of this technique reduces the average word error rate from 42.25% to 35.5% for 4 Indic languages. The proposed model is additionally applied to assess its effectiveness in handling noisy data with colloquial speech, thereby verifying its robustness and applicability in diverse conditions.

CCS Concepts: • **Computing methodologies** → **Speech recognition**.

Additional Key Words and Phrases: Automatic Speech Recognition, Indic Languages, ISFASR, Masked Language Modelling, XLM Roberta, Difflib, Spell correction

ACM Reference Format:
Rajalakshmi R and Sujith M. 2018. Automatic Speech Recognition in Indic Languages with enhanced Spell Checker using Masked Language Modelling. *Proc. ACM Meas. Anal. Comput. Syst.* 37, 4, Article 111 (August 2018), 18 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

In natural language processing, automatic speech recognition (ASR) is a game-changing technology that makes it easier to translate spoken language into written text. Significant advances in machine learning, deep learning, and neural network architectures have propelled the evolution of ASR. Due to this advancement, a wide range of applications, including voice assistants and transcription services have seen widespread adoption.

Certain languages, often referred to as low-resource languages, present particular opportunities and challenges in the field of ASR. Most of the Indic languages, including Tamil, Malayalam, Kannada and Marathi belong to this category. Unlike English, these languages do not have many ASR systems developed to provide state-of-the-art results. Despite the presence of a large speaking population and use cases for these languages, the insufficiency of resources inhibits the achievement of good results for ASR. There are also several vulnerable individuals in society such as the elderly and transgender people. Some people have problems with their speaking in general. This makes it difficult for them to carry out a conversation in public places such as hospitals, restaurants and

Authors' addresses: Rajalakshmi R, Vellore Institute of Technology Chennai, Chennai, India, rajalakshmi.r@vit.ac.in; Sujith M, Vellore Institute of Technology Chennai, Chennai, India, sujith.m2020@vitstudent.ac.in.

ACM 2476-1249/2018/8-ART111
https://doi.org/XXXXXXX.XXXXXXX

railway stations. This bottleneck has been present for a long time and there is no proper working solution to resolve the issue. The development of such a system can help all these people to have a conversation with others with ease. Moreover, a good speech recognition system can lead to the development of a better translation system which can further help to break the language barrier among people.

Over the years, several ASR-based systems have been developed for various languages. The initial models comprised Hidden Markov Models [1], succeeded by RNN-based architectures [14, 15]. Currently, Transformer-based architectures are employed [5]. The Whisper model [16] from OpenAI is the current state-of-the-art in this field. Thus, in this research, we fine-tune the Whisper medium for four different languages. Although the model provides state-of-the-art results, the word and character errors in the transcriptions make it unsuitable for real-world usage. Thus, the errors present in the transcriptions from the speech recognition model can reduced by incorporating two techniques, namely, masked language modelling and word similarity matching.

The main contributions of our work are:

- Proposing a fine-tuned ASR model for four Indic languages - Tamil, Malayalam, Marathi and Kannada. An average word and character error rate of 42.25% and 13.25% is achieved with the test data.
- Introducing an error correction model by integrating two techniques - masked language modelling and word similarity matching. Masked language modelling is employed with the XLM-Roberta and Indic Bert models. The average word error rate is reduced to 35.25% using this approach.
- Cross-data validation of the proposed methodology on colloquial speech data in Tamil spoken by vulnerable individuals to check its performance. The ASR model resulted in the transcriptions with a word error rate of 71%. It is reduced to 68% through error rectification.

The overall content of the paper is structured as follows: a literature survey of the papers related to the field of automatic speech recognition and error correction is presented in section 2, followed by the description and statistics of all the datasets used in section 3. The proposed methodology including speech recognition and error correction is offered in section 4. Section 5 describes the experimental setup in detail including the performance metrics. All the results obtained and details discussions are presented in section 6. The conclusion is provided in section 7.

2 RELATED WORK

2.1 Speech Recognition

The development of ASR systems started with Hidden Markov Models. The performance of the native HMM was overcome by using a hybrid CNN-HMM based model [1] as it incorporates convolutional neural networks. A Gated Recurrent Fusion (GRF) method with joint training framework was proposed for robust end-to-end ASR [9]. The GRF is used to address the speech distortion problem by removing the noise signals. A hybrid RNN-LSTM model [14] is proposed to overcome the challenges of the traditional LSTM by using RNN as a forget gate in the network. This model outperforms most of the other deep learning based models. A meta adversarial learning approach was proposed for multilingual low-resource languages to enable a model's encoder to learn more language independent features [6].

The works in ASR were also adapted for several low-resource languages. A multilingual speech recognition system and spell correction system were designed using RNN-GRU [15]. This model shows improved performance compared to RNN based systems for low-resource languages. Speech enhancement is achieved with multistage self-attentive temporal CNNs [12]. Each stage has a self attentive block followed by temporal convolutional networks. The addition of both the blocks resulted in improved performance of the models. Recent advances in Transformer architectures have also shown significant promise for ASR in Tamil. These models utilize self-attention and cross-attention mechanisms to capture long-range dependencies within speech, leading to improved performance on challenging tasks like noise robustness and speaker adaptation. The wav2vec model [5] from

Facebook was a milestone in the field of ASR. Furthermore, the availability of large multilingual ASR models like Whisper [16] opens up new avenues for Tamil ASR. These models can be fine-tuned on smaller amounts of Tamil speech data and achieve competitive performance.

2.2 Error correction

All current state-of-the-art ASR models still produce several errors while transcribing. The errors can occur either at the word or character level. The word level error occurs when the correct word is replaced with a whole different word with a different meaning, whereas the character level error occurs when the lemma of the word is correct but the actual word is incorrect. Thus, similar to the research and development in the field of ASR, several spell and error correction systems and models have been developed to minimize the errors. Using these tools, the errors in the transcriptions of the ASR model can be reduced further.

[15] makes use of Indic Bert incorporating masked language modelling to replace the incorrect words. The model takes the masked sentence as input and generates suggestions for the masked word. The incorrect word is replaced with the most probabilistic suggestion. This model can be used for this purpose in twelve different languages. A pipeline for error correction in code-mixed data is proposed using a five-phase approach [21]. The phases include the detection of code-mixed data using an English language dictionary, making a set of possible transliterations of the words to the target script, normalization of the transliterations and romanizing the normalized words back to English. This pipeline is implemented for four Indic languages Hindi, Gujarati, Bengali and Tamil.

A sequence-to-sequence model is constructed for automatic spelling correction for low-resource languages, Hindi and Telugu [8]. It is a character-level error rectifier using LSTM and a recurrent network with attention. Correction is claimed to be done with about 90% accuracy. Another paper [13] discusses the task of Chinese spell checking, highlighting its significance in natural language understanding. The paper introduces a confusionset-guided decision network for spoken Chinese spell checking. The model leverages confusion sets to generate candidate sets, allowing it to accurately identify incorrect characters through bidirectional long short-term memory. Extensive experiments on logistics data and the SIGHAN Bake-off dataset demonstrate the model's efficiency and superior performance compared to other models. Additionally, the lightweight nature of the model contributes to effective error correction in scenarios with limited computing resources.

The xlm-roberta-base model incorporating masked language modelling is used as a spell checker in a research for the Tamil language [18]. The authors also propose an algorithm to introduce errors in the data for testing purposes. A word corpus is built by scraping various sources such as Wikipedia and Tamil conversations. The proposed model achieves an accuracy of 91% for error detection. A method to identify hate and offensive content in Tamil is proposed in [17]. The authors try to identify a suitable embedding technique for Tamil text representation by employing different transformers models. An experimental study is also undertaken with various types of classifiers along with the transformer models. The best combination which combines data stemming, embedding with MuRIL and using a majority voting based ensemble technique results in an accuracy of 86% for the detection of offensive content.

A study proposes a hybrid approach [2], melding rule-based and neural network methodologies to proficiently handle various grammar features in the Tamil language. The four-phase implementation described in the paper encompasses aspects such as spell checking, and handling consonant errors, long component errors, and subject–verb agreement errors. Notably, the hybrid model exhibits notable success, achieving a 94% detection rate for consonant errors. Comparative analysis with online tools reveals superior performance, suggesting avenues for enhancing deep learning models' accuracy and exploring additional facets of Tamil grammar in future research. Another article addresses the crucial need for effective spell-checkers in applications such as search engines, information retrieval, and emails, emphasizing the challenges faced by Indian languages like Hindi. It proposes the

Table 1. Statistics of the Data from OpenSLR for Four Languages

Language	Number of transcripts (Female)	Number of transcript (Male)	Total Duration (in hrs)
Tamil	2335	1956	7.08
Malayalam	2103	2023	5.51
Kannada	2186	2214	8.48
Marathi	1569	Not Available	3.02

Table 2. Statistics of the Data from OpenSLR for Four Languages

Gender	Literates	Illiterates	Total
Male	4	9	13
Female	7	24	31
Transgender	3	4	7
Total	14	37	51

HINDIA model [19] introducing a novel approach using a deep-learning method, specifically an attention-based encoder–decoder bidirectional recurrent neural network (BiRNN) with long short-term memory cells. Trained on a sizable dataset, HINDIA demonstrates superior performance in spell error detection and correction for Hindi. Its innovative features, including an attention layer and CBOW word-embedding technique, contribute to its effectiveness, surpassing existing deep-learning-based models for regional languages. The research highlights the infancy stage of Indic language-related studies and suggests future directions, including integration into search engines, email systems, and development of standalone writing assistance systems for Hindi.

3 DATASET DESCRIPTION

The Open Source Multi-speaker Speech Corpora (OpenSLR) [10] dataset is used for this research. This dataset contains speech corpus along with transcriptions for six Indian languages, which include Tamil, Malayalam, Telugu, Kannada, Marathi and Gujarati. About 2000 lines of speech are present in each language for both Male and Female speakers except for Marathi which has data from female speakers only. The exact statistical information about the dataset is given in table 1. Another dataset containing colloquial Tamil speech data [3] spoken by vulnerable (elderly and transgender) individuals in the society is also used to validate the adoption our model for colloquial data. The data was prepared as a part of a shared task in Tamil speech recognition [4]. The data has too much noise making the audio unclear and difficult to interpret. The presence of noise and colloquial terms pose a major challenge for the task of automatic speech recognition. The data has about 1200 transcripts from 7.5 hours of audio. The complete statistics for this dataset are presented in table 2.

4 METHODOLOGY

4.1 Automatic Speech Recognition

The model [16] is pre-trained for speech recognition and translation. The model is trained on 680k hours of labelled data and supports about 99 languages. The model reports good results when tested on the LibriSpeech and common voice datasets. There are several versions of the model such as tiny, small, medium and large, each with a different number of parameters. Each model has its use cases. Only some of these models have multilingual

capabilities. The medium version of the model is used in this research. It has 769M parameters that can be used with limited GPU capabilities.

The ASR model's processor has two key parts which are the feature extractor and the tokenizer. The feature extractor extracts key features from the audio data, such as spectrograms or Mel-frequency cepstral coefficients (MFCCs). These features serve as input to the ASR model. The tokenizer encodes the transcripts into tokens, and pads or truncates them as required. These are then fed to the decoder. It also decodes the final predicted tokens back to text. The decoder IDs can be forcefully set for the recognition of a particular language if needed.

The architecture of the pipeline for model training is depicted in figure 1. As the model only accepts audio with a sampling rate of 16KHz, resampling of all the available audio is done using the Librosa package in Python. The MFCCs extracted with the processor are the inputs provided to the encoder blocks of the model. A cross-attention mechanism exists between the encoder and decoder to allow the decoder blocks to attend to the outputs from every encoder block. At the end, the decoder block provides the final output tokens which are decoded by the tokenizer.

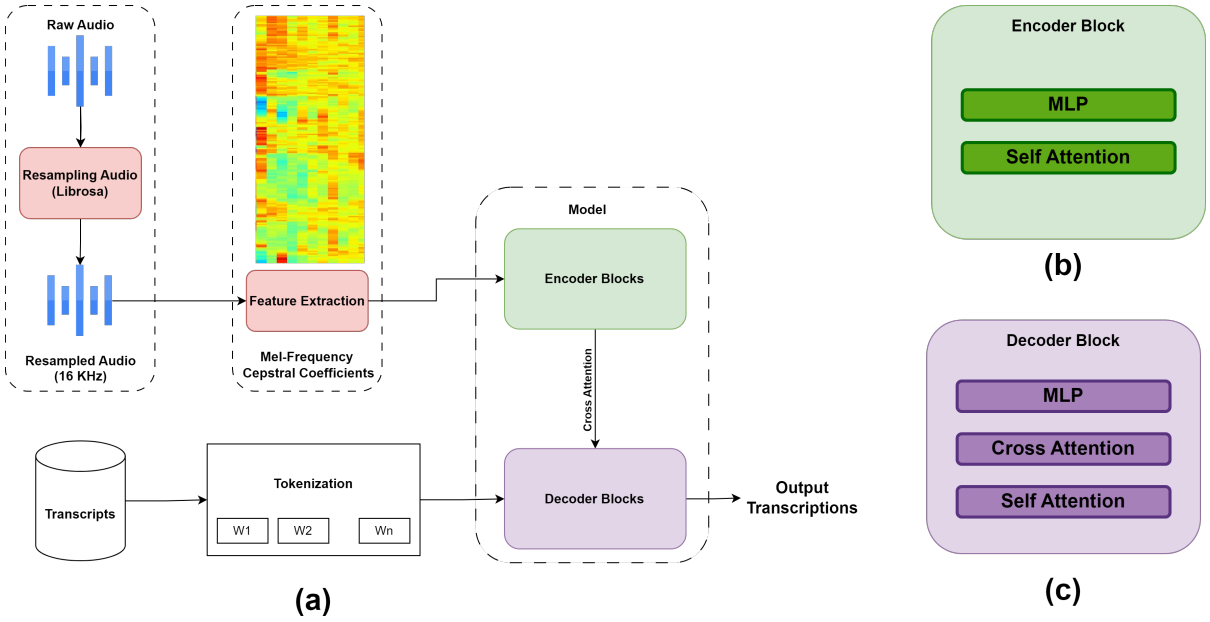


Fig. 1. (a): The architecture of the fine-tuning process for automatic speech recognition. (b): Illustration of the encoder block containing multilayer perceptron and self-attention layers. (c): Illustration of the decoder block containing an extra layer for cross attention apart from the self-attention and MLP layers.

4.2 Spell Correction using masked language modelling and word similarity

As mentioned before, both masked language modelling and word matching techniques are incorporated to correct the errors in the predictions from the ASR model. The architecture of the error correction pipeline is presented in figure 2.

Initially, the predictions from the ASR model are tokenized into words using the NLTK library in Python. Then each word is sent to the word checker to find if it is misspelled. The checker ignores all the punctuation, special characters and characters from a language other than the target language. If the word belongs to the target

language, then the checker tries to find it in the corpus. If found, the word is spelt correctly and no changes are done and the word is appended to the original sentence as it is. If not, then the word is considered to be erroneous.

The misspelled word is now fed to the 'get_closest_matches' function in the Difflib. This function checks for the closest matches of the target word in the corpus and provides 5 unique closest matches as output. Simultaneously, the misspelled word is replaced with the '<mask>' token in the actual sentence and the masked sentence is given as input to the XLM-Roberta model. This model also now generates 5 suggestions for the masked word.

Now, we compute the cosine similarity $\cos\theta$ between every pair of words from the two suggestion lists using the formula in equation 1, where A and B are the word embeddings of the words. The dot product of the two vectors is divided by the product of their lengths. The pair which is the most similar is considered to be the final pair and the word from the Difflib suggested list that belongs to this pair is the final replacement for the misspelt word.

$$\cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} \tag{1}$$

The Roberta model's suggestions are based on the context of the actual sentence whereas the suggestions of the Difflib library are based on the misspelt word alone. Thus, calculating the similarity between the words from the two lists helps us find the word specific to the context of the sentence. The XLM-Roberta model can be replaced with other models that employ masked language modelling. We have repeated the spell check experimentation with the Indic Bert model to get comparative results.

5 EXPERIMENTAL SETUP

This section explains all the implementation details of the experiment including the hardware and software used, the cloud platforms used for training and the performance metrics used for the evaluation of the proposed models.

5.1 Experimental platform

All experiments were conducted on the Kaggle kernel with the GPU P100 accelerator. Python version 3.10.12 was used. The transformers library from hugging-face provides all the resources required for tokenization, feature extraction and training of the model. Librosa library is used for loading the audio files and resampling them. The Evaluate and Jiwer libraries are used for computing word and character error rates. Difflib is used to find the closest matches for the misspelled words from the corpus. Fasttext library in Python is used to obtain the word embeddings needed for computing the similarity. The final cosine similarity value is calculated with Numpy.

The dataset of every language is split into 50%, 25% and 25% for training, validation and testing respectively. The batch sizes for both training and validation were set to '1' due to GPU limitations. The model was trained for 150 steps in each language. The total time taken for training and validation was 10 hours. The learning rate used was 1×10^{-5}

5.2 Performance Metrics

Word Error Rate (WER) is a metric used to evaluate the accuracy of automatic speech recognition (ASR) systems by measuring the difference between the transcriptions generated by the system and the reference transcriptions. It quantifies the percentage of words that are incorrectly transcribed, making it a crucial measure for assessing the overall performance of ASR models. The formula for calculating WER involves three components: the total number of substitutions (S), deletions (D), and insertions (I) needed to transform the system's output into the reference transcription, divided by the total number of words in the reference transcription. The formula is expressed in equation (2), where N represents the total number of words in the reference transcription. Minimizing the WER is a key objective in improving the accuracy of ASR systems, as it directly reflects the extent of discrepancies

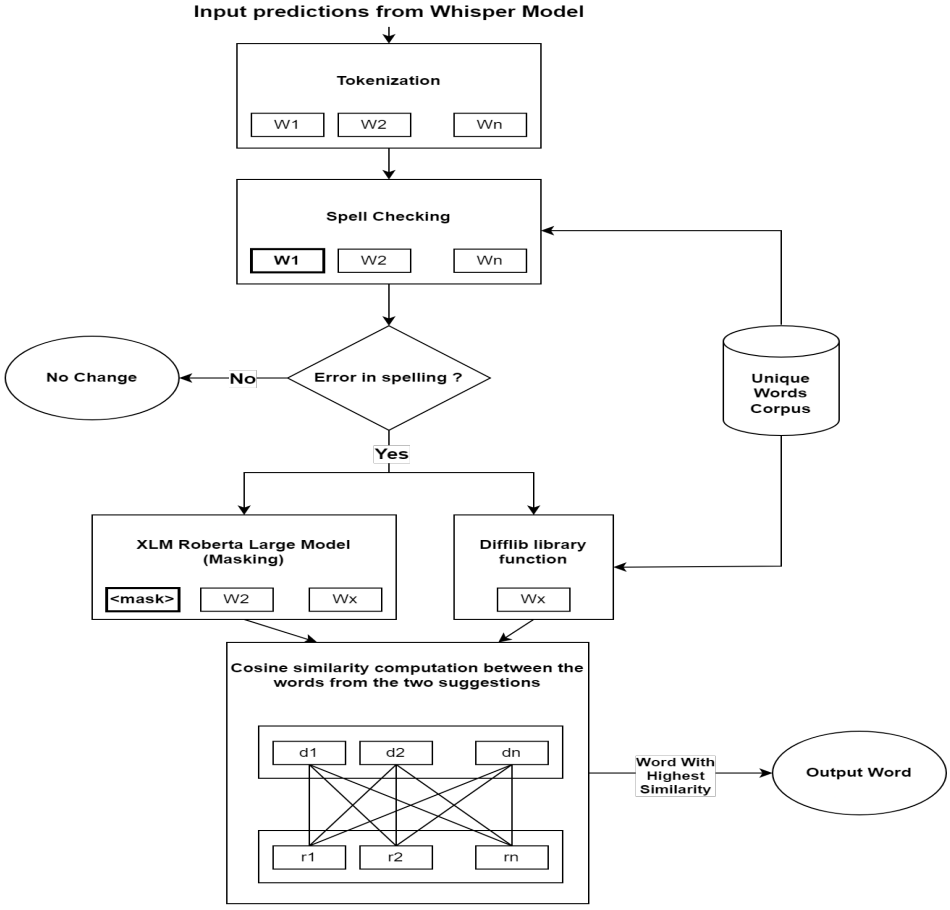


Fig. 2. Architecture of the proposed error correction pipeline

between the generated and reference transcriptions. The same formula can be used for the calculation of character error rate. It calculates the error for each individual characters.

$$WER = \frac{S + D + I}{N} \tag{2}$$

Another metric, the sequence match ratio is also computed between the actual and the predicted transcriptions using the formula in equation 3. This ratio is used to portray the similarity between two sentences and is also known as similarity ratio.

$$SequenceMatchRatio(sentence1, sentence2) = \frac{No.of MatchingUnigrams}{\max(len(sentence1), len(sentence2))} \times 100 \tag{3}$$

6 RESULTS AND DISCUSSION

The proposed speech recognition and error correction architecture yields good results and performs better than the existing state-of-the-art models available. In this section, the proposed model is compared with other models to demonstrate the reduction in the error achieved through our model.

This section is divided into three subsections, subsection 6.1 showcases and discusses the main results obtained from the fine tuning the ASR model and the results after adding the spell correction model to them. Section 6.2 is the comparative study where our proposed work is compared with the other existing works. Section 6.3 is an ablation study that discusses the significance of various components of our architecture and the impact of our model on different versions of a language.

6.1 Experimental Result

The ASR model was successfully fine-tuned on the OpenSLR and Colloquial Tamil speech datasets for four Indic languages in total. The reduction in the word error rate of the training dataset during the fine tuning process of our ASR model is illustrated in figure 3. Marathi had the least word error rate at the start and the end due to its relatively smaller dataset size which is almost half the size of the other datasets. The model's performance in Malayalam is not on par with its performance in other languages. This is mainly due to the training size of the audio data used in the actual pretrained whisper model for the Malayalam language which is lower than the training size of the rest of the languages.

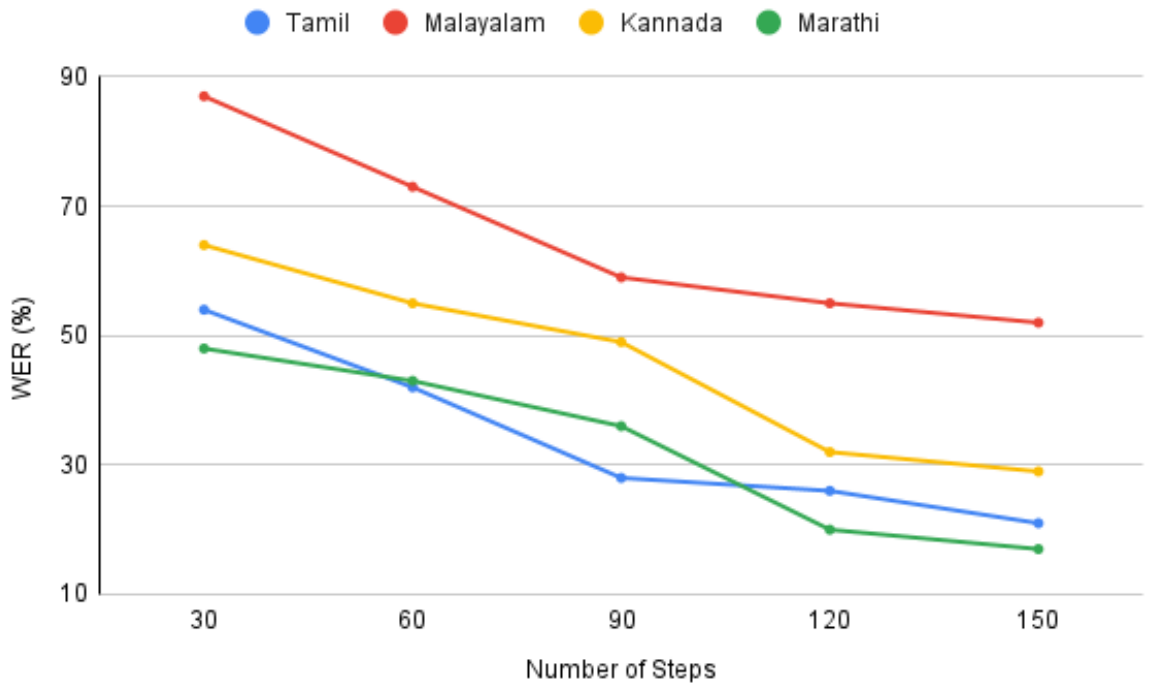


Fig. 3. Changes in the word error rate in % for train dataset with the number of epochs during the training of the ASR model

Table 3. Word Error Rate for the results obtained from the fine-tuned ASR model for train, validation and test data

Language or Dataset	Train Dataset	Validation Dataset	Test Dataset
Tamil (SLR 65)	0.21	0.35	0.36
Malayalam (SLR 63)	0.51	0.64	0.66
Kannada (SLR 79)	0.28	0.41	0.47
Marathi (SLR 64)	0.18	0.21	0.20
Colloquial Tamil Dataset	0.60	0.64	0.71

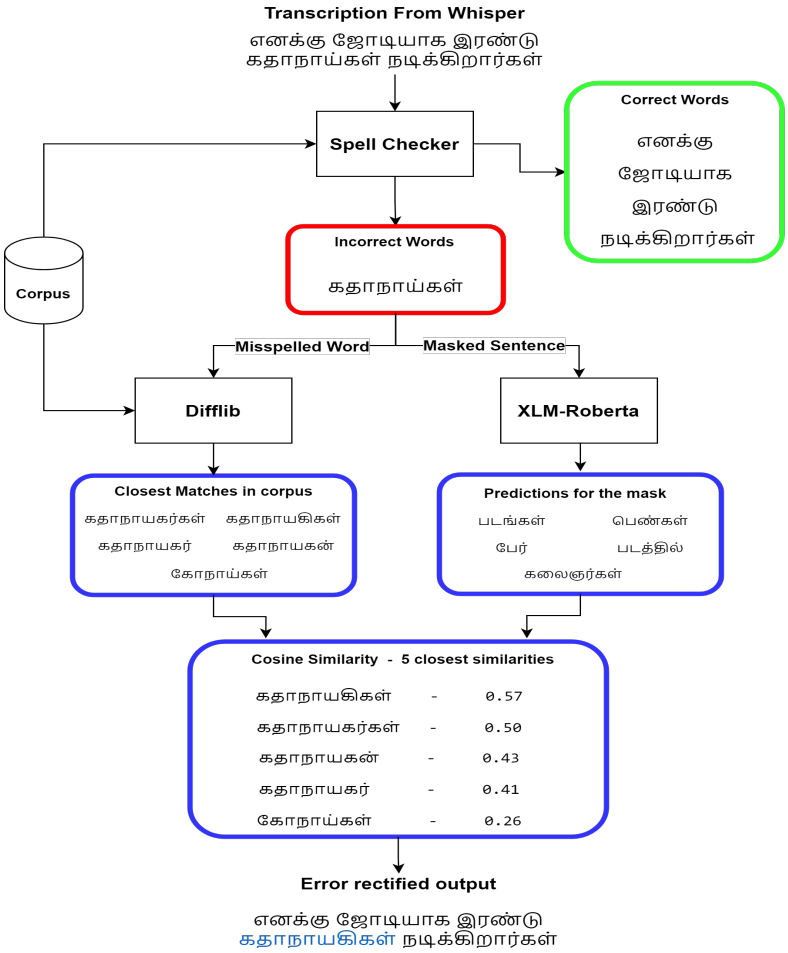


Fig. 4. Illustration of the working of the error correction model with an example

The word error rate of the transcriptions from the speech recognition model for train, validation and test data of the OpenSLR dataset is given in table 3. The average word error rates for the test datasets is 42.25%. Although the error rate for Marathi is the lowest, given the size of the test data (1000 for Tamil and 400 for Marathi), the

Table 4. Word error rate in % for the test dataset before and after the addition of the spell correction model with and without xlm-roberta

Language	Tamil(slr 65)	Malayalam(slr 63)	Kannada(slr 79)	Marathi(slr 64)
Fine tuned ASR model	36	66	47	20
Fine tuned ASR model+Difflib	31	60	41	18
Fine tuned ASR model+Difflib+Roberta	**29**	**57**	**38**	**18**

performance of the model in Tamil can be considered the best. The error rates for colloquial Tamil dataset is also included. The respective error rate is much higher than those for pure languages. This shows that the proposed ISFASR model faces difficulty in recognizing colloquial speeches.

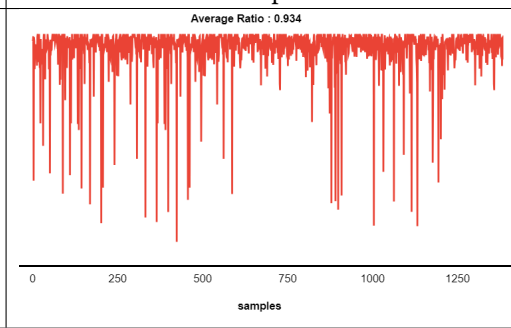
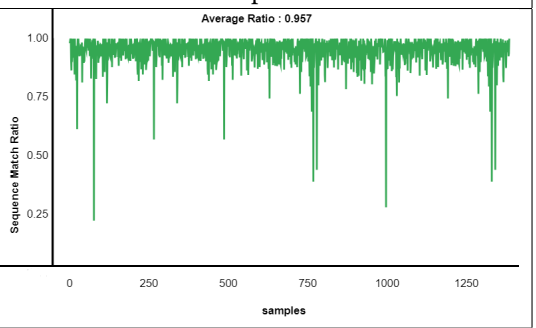
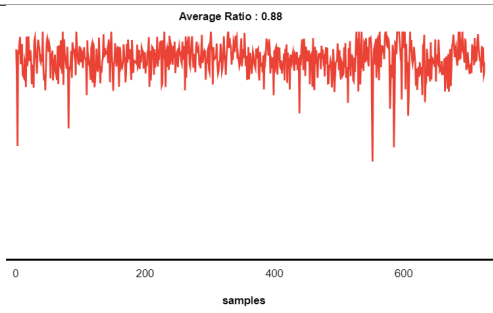
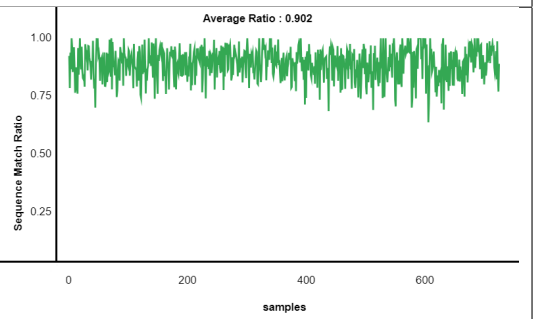
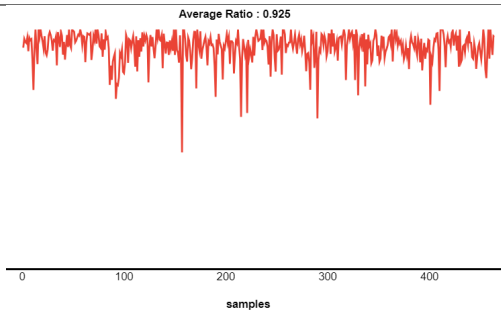
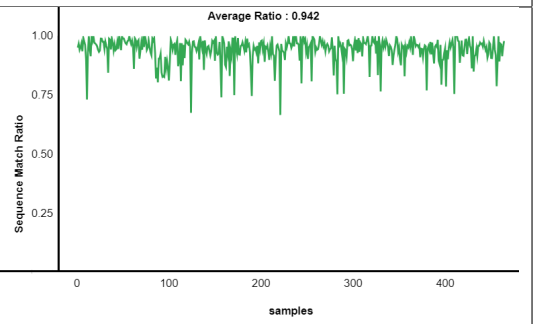
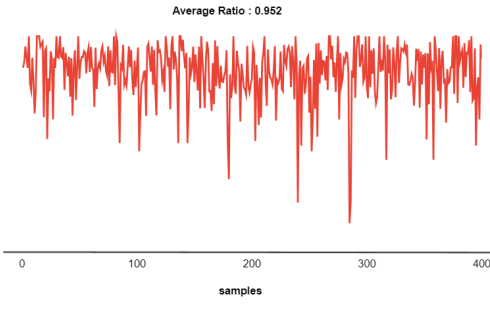
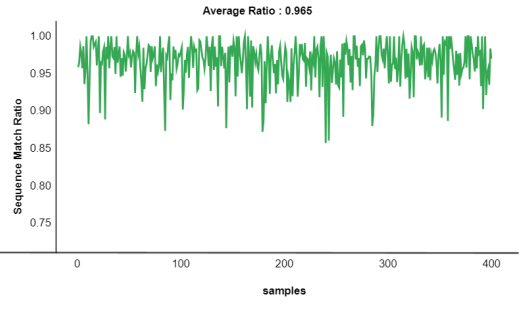
After the ASR model, the transcriptions are sent to the error correction model. The complete working of the process is illustrated in figure 4 with an example from the Tamil language . As depicted in the figure, four of five words in the given transcription are correct. Thus, they are left unchanged. The model now only works with the one misspelled word. The word is sent to the Difflib library function. The transcription is masked for the respective misspelt word and sent to the xlm-roberta model. Both the model and the function provide five suggestions each for the misspelt word. Now, the cosine similarity computation based on equation 1 is performed for every pair between the suggestions and an average is computed for every word suggested by the Difflib library function. The average similarity values are displayed along with the respective words in the figure. The word with the highest average similarity is considered to be the final replacement for the misspelt word in the transcription.

A few sample results of the transcriptions and the respective error corrections of the OpenSLR data are presented in figure 5. Similarly, figure 6 shows the results for colloquial Tamil speech data. For the error correction of the colloquial speech, the MLM model was neglected and only the Difflib library function is used. All possible colloquial Tamil lexicons were gathered from various sources and were added to the main corpus.

Table 4 shows the results of the model with and without the addition of the proposed error correction architecture. The word error rate reduces by about 5% when the error correction is done with the Difflib library function alone without a maked language model. It further reduces by another 2% with the addition of xlm-roberta taking the final average word error rate to 32.25%. The change in the individual error rate is proportional to the initial error rate. The change in error rate is higher for Malayalam and Kannada as their initial error rate was higher than the rest of them. The reduction in the WER for Marathi is the lowest. Unlike the rest of languages, the addition of the xlm-roberta model does not have an impact on the Marathi language. Similarly, the changes in the character error rate with the introduction of the spell corrector is portrayed in figure 7. The reduction in the character error rate for Tamil and Marathi are analogous. However, for Malayalam, the change in the CER is very small. Such a small change in the character error rate leads to a bigger impact on WER. This implies, for Malayalam, most of the errors were at the character level rather than the word level. Due to this, the character error rate for Kannada is higher than that of Malayalam, whereas it is vice versa for the word error rate of the two languages.

The sequence match ratio was computed between the predictions and the actual transcriptions before and after the usage of a spell checker. The graphs of the sequence match ratio for all the samples in the test data is depicted in table 6 along with the average ratio. The reduction in the height of the spikes (lowest point in graph) implies the increase in the ratio's value. On an average, the lines in the graph have moved up for all the four languages.

Table 5. Sequence match ratio of the test data before and after the addition of spell checker for all four languages

Language	SMR before spell correction	SMR after spell correction
Tamil		
Malayalam		
Kannada		
Marathi		

Language	True Transcripts	Transcriptions from Whisper	Error rectified transcripts
Tamil	எனக்கு ஜோடியாக இரண்டு கதாநாயகிகள் நடிக்கிறார்கள்	எனக்கு ஜோடியாக இரண்டு கதாநாய்கள் நடிக்கிறார்கள்	எனக்கு ஜோடியாக இரண்டு கதாநாயகிகள் நடிக்கிறார்கள்
Malayalam	അദ്ദേഹത്തിനെ പറ്റിയുള്ള കൂടുതൽ വിവരങ്ങൾ ലഭ്യമല്ല	അദ്ദേഹത്തിനെ പറ്റിയുള്ള കൂടുതിൽ വിവരങ്ങൾ ലഭയമല്ല	അദ്ദേഹത്തിനെ പറ്റിയുള്ള കൂടുതൽ വിവരങ്ങൾ ലഭ്യമല്ല
Kannada	ನೀರಿನಲ್ಲಿ ತೇಲುವಾಗ ಮೇಲ್ಭಾಗದಲ್ಲಿರುವ ಕಪ್ಪು ಛಾಯೆಯ ಭಾಗ ಅದಕ್ಕೆ ಪ್ಯಾಣಿಧ್ಯವ ಎಂದು ಹೆಸರು	ನೀರಿನಲ್ಲಿ ತೇಲುವಾಗ ಮೇಲ್ಭಾಗದಲ್ಲಿರುವ ಕಪ್ಪು ಚಾಯಯ ಭಾಗ ಅದಕ್ಕೆ ಪ್ಯಾಣಿದ್ಯುವ ಎಂದು ಹೆಸರು	ನೀರಿನಲ್ಲಿ ತೇಲುವಾಗ ಮೇಲ್ಭಾಗದಲ್ಲಿರುವ ಕಪ್ಪು ಚಾಯಾಪಚಯ ಭಾಗ ಅದಕ್ಕೆ ಪ್ಯಾಣಿಧ್ಯವ ಎಂದು ಹೆಸರು
Marathi	काही लोक कुबेराची पूजा करतात तर काही लोक देवीची पूजा करून तूपभात व साखर खातात	काही लोक कुबेराची पूजा करतात तर काही लोक देवाची पूजा करून रुपात् साखर खातात	काही लोक कुबेराची पूजा करतात तर काही लोक देवाची पूजा करून तूपभात साखर खातात

Fig. 5. Sample transcriptions resulted from the ASR model along with the errors rectified using spell checker on OpenSLR dataset

Table 6. Comparison of the word error rate for our proposed work with other existing works

Language	Tamil	Malayalam	Kannada	Marathi
GRU [15]	0.84	0.87	0.83	0.65
GRU + Indic Bert [15]	0.73	0.78	0.75	0.53
Proposed ISFASR	**0.29**	**0.57**	**0.37**	**0.18**

6.2 Comparative Study

The OpenSLR dataset used for our research has also been used by other researches working on the field of automatic speech recognition. In this section, we compare the performance of our model with other existing works making use of the same dataset. As mentioned in section 2, a research has been done with this dataset for speech recognition using RNN-GRU model and error correction using Indic Bert. Table 6 gives the results of comparison for all the four languages. Our proposed model performs better than their work in all the four

sno	True Transcripts	Transcriptions from Whisper	Error rectified transcripts
1	அப்டித்தாங்க எல்லாரும் நம்மல கண்டுபுடுச்சு போறாங்க	அப்படிதாங்க எல்லாரும் நம்மல கண்டுபிடிச்சுட்டு போயிறாங்க	அப்படியாங்க எல்லாரும் நிம்மல கண்டுபிடிச்சுட்டு போறாங்க
2	போகும் போது சண்ட இழுக்குறாங்களா	போகும்போது சண்ட எழுக்குறாங்களா	போகும் போது சண்ட இழுக்குறாங்களா
3	நம்மளுக்குன்னு ஒரு உறுத்து இருக்குலா நம்மளுக்குன்னு ஒரு தோரண	நம்முளுக்குன்னு ஒரு உருத்து இருக்குல நம்முளுக்குன்னு ஒரு தோர்டன	நம்மளுக்குன்னு ஒரு உருத்து இருக்குலா நம்மளுக்குன்னு ஒரு தோரண

Fig. 6. Sample transcriptions resulted from the ASR model along with the errors rectified using spell checker on colloquial Tamil dataset

Table 7. Comparison of the character error rate in % for our proposed work with other existing works

Language	Malayalam	Marathi	Kannada	Tamil
MTL-ASR [6]	80.76	87.13	-	-
MML-ASR [6]	42.56	37.87	-	-
MADML-ASR [6]	41.62	35.93	-	-
Fine tuned ASR model	**15.82**	**6.9**	**19.37**	**10.28**
Proposed ISFASR	**15.37**	**4.52**	**10.9**	**7.86**

languages. The difference in the average word error rate is about 28%. In fact, their paper only reports the results for their validation dataset. Thus, our test results being much better than their validation results proves the superiority of our proposed model.

Another research proposed a meta adversarial learning approach for speech recognition in low resource languages. This paper has used the OpenSLR dataset for Malayalam and Marathi languages. The paper makes use of the character error rate metric to report the results. The comparison of our model with the works in that paper

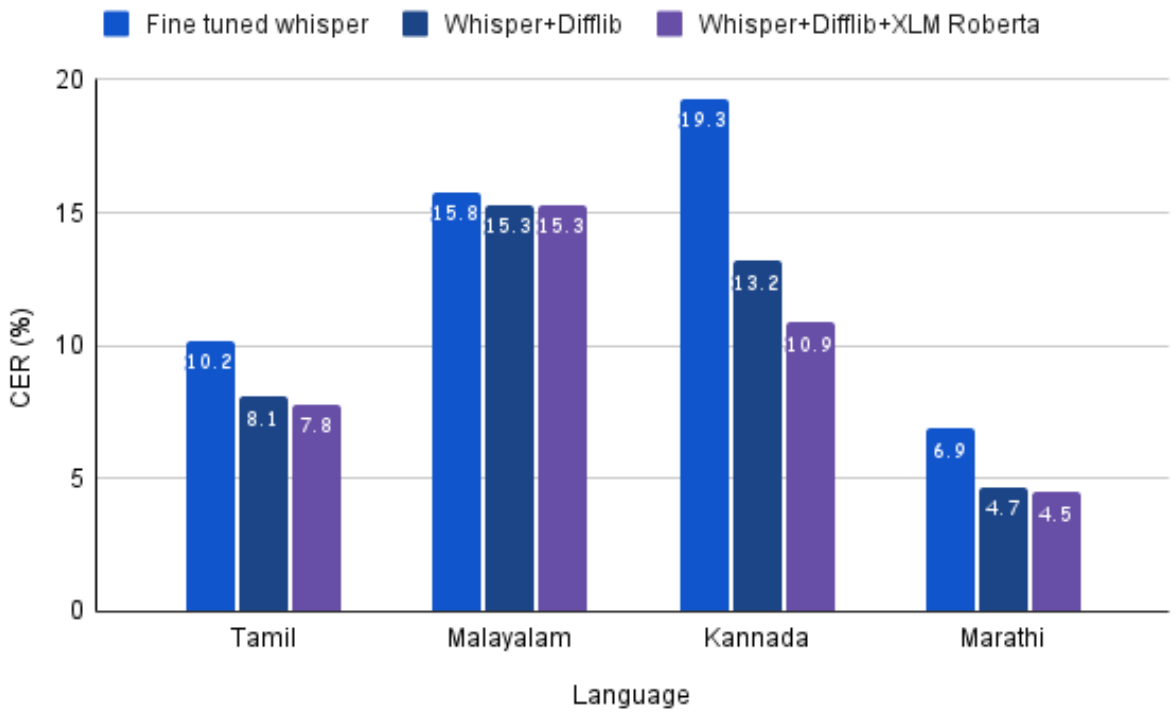


Fig. 7. Reduction in the character error rate in % for the test dataset before and after the addition of the spell correction model with and without xlm-roberta

Table 8. Changes in the word error rate for using different masked language models in the spell checker

Language	Tamil	Malayalam	Kannada	Marathi
No MLM	31.26	60.8	41.11	18.84
Indic Bert	31.08	59.91	40.89	19.87
multilingual Bert	30.19	59.07	40.38	18.86
XLM Roberta	**29.17**	**57.33**	**37.73**	**18.31**

is provided in table 7. A fine tuned ASR model achieves better performance their proposed works. Including the spell checker reduces the error rate further and provides a difference of 28% in error rate between their work and our work.

6.3 Ablation Study

6.3.1 Impact of integrating a masked language model with Difflib library function. Although the Difflib library provides many closest matches for the incorrect words, they are solely based on the edit distance between the target word and the suggestions. It is the masked language modelling technique that captures the context of the transcriptions to provide suggestions for the masked/ misspelt word. There are several masked language models

available supporting multiple languages such as multilingual-Bert [7], xlm-roberta, MP_net [20], xlm-mlm-100-1280 [11] and Indic Bert.

We experimented with integrating the Roberta, multilingual Bert and Indic Bert models for error correction. Table 8 shows the changes in the word error rate in all four languages for different experiments conducted. However, not all models provide good performance in this method. While using the Indic Bert model, the change in the error rate is insignificant. For Marathi, the error rate increases by one percentage making it unsuitable for the language. Multilingual Bert reduces the error rate for Tamil, Malayalam and Marathi by one percentage. This model also does not have an impact on the Marathi language. On the other hand, using the XLM-Roberta model along with the library function reduces the average error rate by about 2-3 percentage all languages expect for Marathi. This implies the importance of capturing the context of a sentence by incorporating masked language modelling. However, as seen in the case of Marathi language, all the three models have the capacity to reduce the error until a particular percentage after which it becomes difficult to correct the errors. However, as observed with the Marathi language, all three models exhibit the capability to reduce errors up to a certain threshold percentage, beyond which error correction becomes challenging.

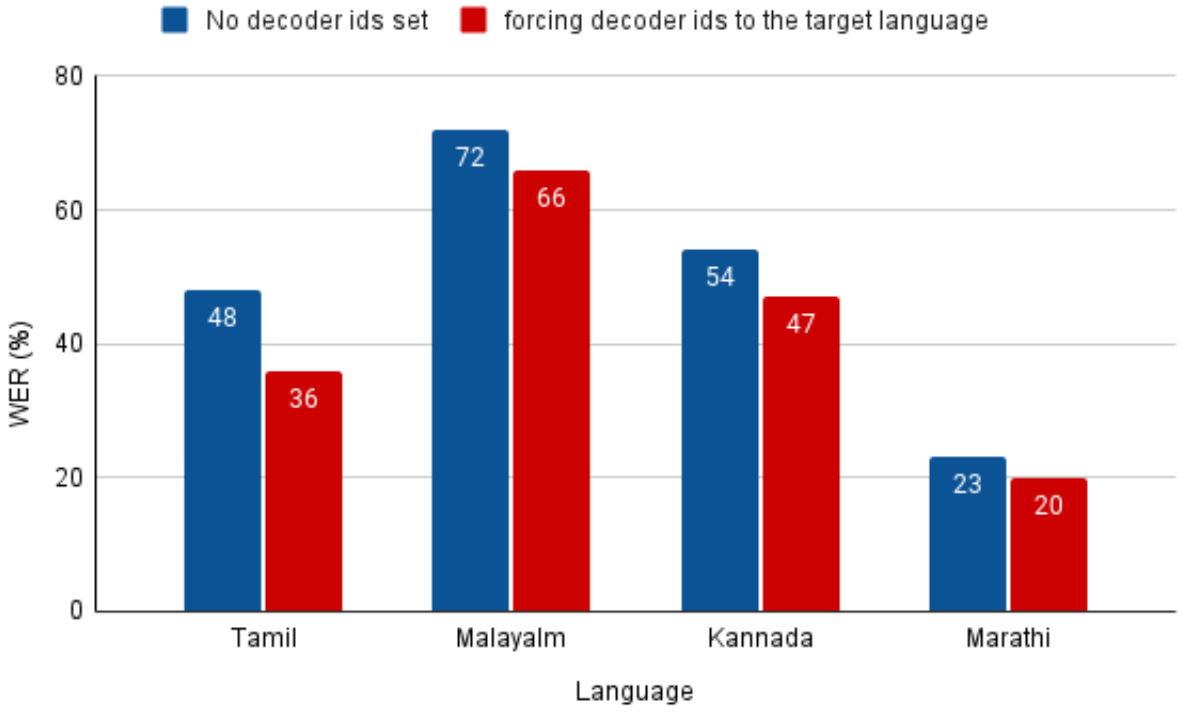


Fig. 8. Comparison of the ASR model's performance with and without setting the decoder ids

6.3.2 Impact of forcibly setting the decoder ids to the target language. Every multilingual speech recognition model has a specific set of decoder ids assigned for every language it supports. Thus, the same token id is not used for two different languages. The models identify the language of the speech and transcribes the speech data using designated decoder IDs. We can forcibly set the decoder ids to a particular language during the training of the model. However, once set, the model can be used for this task in a single language only and cannot be

multilingual. All audio data belonging to a language other than the target language may result in erroneous output transcriptions. We trained four individual models for each language by forcibly setting the decoder ids. Simultaneously, we trained a single model for all four languages with no decoder ids specified. The same training data was used for every languages. Figure 8 shows the performance of the two types of models for the test data of the respective languages. The word error rate is reduced by 7% when the models are training individually by forcing the decoder ids. Thus, if we need to achieve better performance despite the utilisation of more resources, then forcing the decoder ids is an ideal choice.

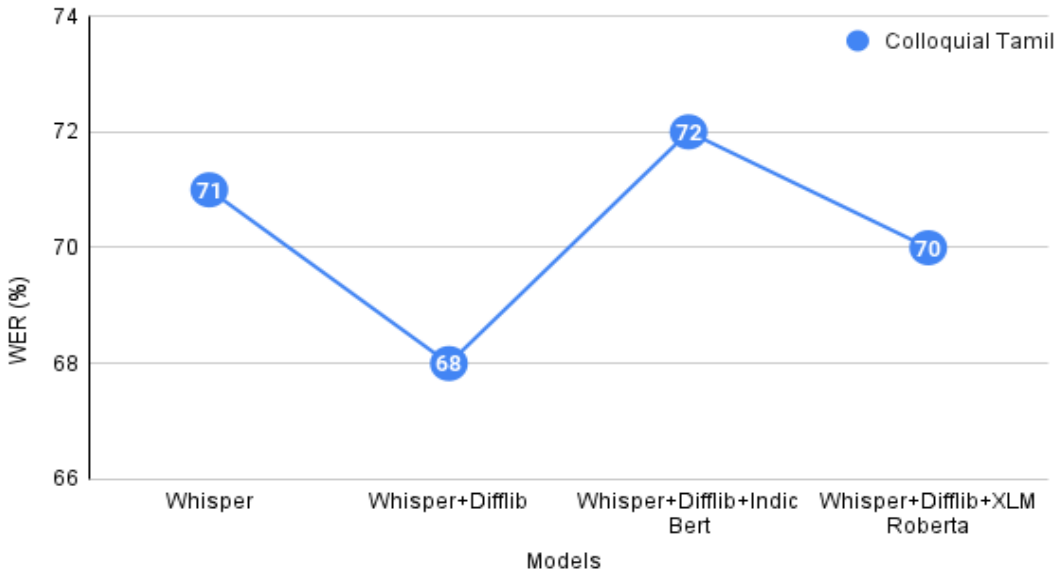


Fig. 9. Changes in the WER for colloquial Tamil speech data with different approaches proposed

6.3.3 Impact of the proposed model on colloquial speech data. Although a colloquial language may appear to be similar to the normal version of the language, there are too many spelling variations between them. The changes between the versions are noticeable in text form rather than during a speech. Thus, a model fine-tuned with lexicons belonging to the pure version of a language cannot be expected to perform for its colloquial form. Due to this, the speech recognition model gives an average performance with colloquial speech data.

Most of the errors occur due to the differences in the spelling of words between these two versions. We experimented with the spell correction model for this data by adding in all possible colloquial lexicons of the Tamil language we could gather from various sources. The changes in the error rate for this dataset are depicted in figure 9.

Replacing the misspelt words with the closest matches works for this data too. However, the integration of a masked language model is not suitable for this type of data as the model was pretrained with a different version of the lexicons, it could not capture the context of the sentence and the suggestions cannot be of the same form.

7 CONCLUSION AND FUTURE WORK

In this research, we have focused on advancing automatic speech recognition (ASR) capabilities for low-resource Indic languages such as Tamil, Malayalam, Marathi, and Kannada. Leveraging the Whisper-medium pre-trained

ASR model from OpenAI, we fine-tuned it and rigorously evaluated its performance using metrics such as word and character error rates. Our study showcases the efficacy of our fine-tuned model in real-world applications like transcription services and voice-driven technologies. To minimize prediction errors, we integrated the XLM-Roberta-Large model for masked language modelling and utilized the Difflib library for word matching, resulting in a 7% reduction in the average word error rate across the four Indic languages. The IndicSpellFixASR (ISFASR) model proposed in this research lays a foundation for improved accessibility and usability of ASR technology, particularly for languages with limited linguistic resources, potentially benefiting various user groups, including the elderly, transgender individuals, and those with speech impairments.

Furthermore, our model's robustness in handling noisy data and colloquial speech was demonstrated. Although the integration of a masked language model does not work for this data, in future, a model that can be specifically pretrained with a colloquial corpus could make a significant impact on colloquial language speech recognition, underscoring its versatility and applicability across diverse linguistic contexts.

REFERENCES

[1] Ossama Abdel-Hamid, Abdel-rahman Mohamed, Hui Jiang, Li Deng, Gerald Penn, and Dong Yu. 2014. Convolutional Neural Networks for Speech Recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 22, 10 (2014), 1533–1545. https://doi.org/10.1109/TASLP.2014.2339736

[2] S. Anbukkarasi and S. Varadhaganapathy. 2022. Neural network-based error handler in natural language processing. *Neural Computing and Applications* 34, 23 (01 Dec 2022), 20629–20638. https://doi.org/10.1007/s00521-022-07489-7

[3] Bharathi B, Bharathi Raja Chakravarthi, Subalalitha Cn, Sripriya N, Arunaggiri Pandian, and Swetha Valli. 2022. Findings of the Shared Task on Speech Recognition for Vulnerable Individuals in Tamil. In *Proceedings of the Second Workshop on Language Technology for Equality, Diversity and Inclusion*. Association for Computational Linguistics, Dublin, Ireland, 339–345. https://doi.org/10.18653/v1/2022.ltedi-1.52

[4] Bharathi B, Bharathi Raja Chakravarthi, Sripirya N, Rajeswari Natarajan, Suhasini S, and Swetha Valli. 2024. Overview of the Third Shared Task on Speech Recognition for Vulnerable Individuals in Tamil. In *Proceedings of the Fourth Workshop on Language Technology for Equality, Diversity, Inclusion*. European Chapter of the Association for Computational Linguistics, Malta.

[5] Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations. arXiv:2006.11477 [cs.CL]

[6] Yaqi Chen, Xukui Yang, Hao Zhang, Wenlin Zhang, Dan Qu, and Cong Chen. 2024. Meta adversarial learning improves low-resource speech recognition. *Computer Speech Language* 84 (2024), 101576. https://doi.org/10.1016/j.csl.2023.101576

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv:1810.04805 [cs.CL]

[8] Pravallika Etoori, Manoj Chinnakotla, and Radhika Mamidi. 2018. Automatic Spelling Correction for Resource-Scarce Languages using Deep Learning. In *Proceedings of ACL 2018, Student Research Workshop*, Vered Shwartz, Jeniya Tabassum, Rob Voigt, Wanxiang Che, Marie-Catherine de Marneffe, and Malvina Nissim (Eds.). Association for Computational Linguistics, Melbourne, Australia, 146–152. https://doi.org/10.18653/v1/P18-3021

[9] Cunhang Fan, Jiangyan Yi, Jianhua Tao, Zhengkun Tian, Bin Liu, and Zhengqi Wen. 2020. Gated Recurrent Fusion with Joint Training Framework for Robust End-to-End Speech Recognition. arXiv:2011.04249 [cs.SD]

[10] Fei He, Shan-Hui Cathy Chu, Oddur Kjartansson, Clara Rivera, Anna Katanova, Alexander Gutkin, Isin Demirsahin, Cibu Johny, Martin Jansche, Supheakmungkol Sarin, and Knot Pipatsrisawat. 2020. Open-source Multi-speaker Speech Corpora for Building Gujarati, Kannada, Malayalam, Marathi, Tamil and Telugu Speech Synthesis Systems. In *Proceedings of The 12th Language Resources and Evaluation Conference (LREC)*. European Language Resources Association (ELRA), Marseille, France, 6494–6503. https://www.aclweb.org/anthology/2020.lrec-1.800

[11] Guillaume Lample and Alexis Conneau. 2019. Cross-lingual Language Model Pretraining. arXiv:1901.07291 [cs.CL]

[12] Ju Lin, Adriaan J. de Lind van Wijngaarden, Kuang-Ching Wang, and Melissa C. Smith. 2021. Speech Enhancement Using Multi-Stage Self-Attentive Temporal Convolutional Networks. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 29 (2021), 3440–3450. https://doi.org/10.1109/TASLP.2021.3125143

[13] Chuanshuai Ma, Miao Hu, Junjie Peng, Cangzhi Zheng, and Qianqian Xu. 2023. Improving Chinese spell checking with bidirectional LSTMs and confusionset-based decision network. *Neural Computing and Applications* 35, 21 (01 Jul 2023), 15679–15692. https://doi.org/10.1007/s00521-023-08570-5

[14] Jane Oruh, Serestina Viriri, and Adekanmi Adegun. 2022. Long Short-Term Memory Recurrent Neural Network for Automatic Speech Recognition. *IEEE Access* 10 (2022), 30069–30079. https://doi.org/10.1109/ACCESS.2022.3159339

[15] M. C. Shunmuga Priya, D. Karthika Renuka, L. Ashok Kumar, and S. Lovelyn Rose. 2022. Multilingual low resource Indian language speech recognition and spell correction using Indic BERT. *Sādhanā* 47, 4 (05 Nov 2022), 227. https://doi.org/10.1007/s12046-022-01973-5

[16] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2022. Robust Speech Recognition via Large-Scale Weak Supervision. arXiv:2212.04356 [eess.AS]

[17] Ratnavel Rajalakshmi, Srivarshan Selvaraj, Faerie Mattins R., Pavitra Vasudevan, and Anand Kumar M. 2023. HOTTEST: Hate and Offensive content identification in Tamil using Transformers and Enhanced STemming. *Computer Speech Language* 78 (2023), 101464. https://doi.org/10.1016/j.csl.2022.101464

[18] Ratnavel Rajalakshmi, Varsha Sharma, and Anand Kumar M. 2023. Context Sensitive Tamil Language Spellchecker Using RoBERTa. In *Speech and Language Technologies for Low-Resource Languages*, Anand Kumar M, Bharathi Raja Chakravarthi, Bharathi B, Colm O'Riordan, Hema Murthy, Thenmozhi Durairaj, and Thomas Mandl (Eds.). Springer International Publishing, Cham, 51–61.

[19] Shashank Singh and Shailendra Singh. 2021. HINDIA: a deep-learning-based model for spell-checking of Hindi language. *Neural Computing and Applications* 33, 8 (01 Apr 2021), 3825–3840. https://doi.org/10.1007/s00521-020-05207-9

[20] Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. MPNet: Masked and Permuted Pre-training for Language Understanding. arXiv:2004.09297 [cs.CL]

[21] Krishna Yadav, Md Akhtar, and Tanmoy Chakraborty. 2022. Normalization of Spelling Variations in Code-Mixed Data. In *Proceedings of the 19th International Conference on Natural Language Processing (ICON)*, Md. Shad Akhtar and Tanmoy Chakraborty (Eds.). Association for Computational Linguistics, New Delhi, India, 269–279. https://aclanthology.org/2022.icon-main.33