sciendo INTERNATIONAL JOURNAL ON SMART SENSING AND INTELLIGENT SYSTEMS

Research Article | DOI: 10.2478/ijssis-2025-0009 Issue 1 | Vol. 18 (2025)

Enhanced lstm network with semi-supervised learning and data augmentation for low-resource ASR

Tripti Choudhary*, Vishal Goyal and Atul Bansal

Department of Electronics and Communication, GLA University, Mathura, India

*E-mail: triptichoudhary06@gmail.com

Received for publication November 20, 2024.

Abstract

Automatic speech recognition (ASR) is essential for developing intelligent systems capable of accurately processing human speech, particularly in low-resource languages. This study addresses the challenges faced by ASR systems in Indian languages, where data and resources are limited. The authors propose a novel three-step methodology that combines data augmentation and semi-supervised learning to enhance ASR performance. First, an enhanced long short-term memory (LSTM) network is used to train a baseline model with limited labeled data. Next, synthetic data is generated and combined with original recordings to refine the ASR model. Finally, semi-supervised training further boosts accuracy. Evaluations demonstrate significant improvements over existing models for Hindi, Marathi, and Odia languages.

Keywords

Automatic Speech Recognition, Data Augmentation, Semi-supervised learning, Low-resource ASR

I. Introduction

Due to its importance as a means of human communication, automatic speech recognition (ASR) has attracted more attention from scientists in recent decades. Starting with small ASR models that only recognized a limited set of sounds, ASR has progressed into complex ones that can react naturally to diverse language sounds. There has been growing interest in ASR technology because of the need to automate low-level operations that require contact between humans and machines. The variety of human speech makes automatic recognition a challenging task. ASR is being used in broader contexts, including weather forecasting, automated phone systems, stock price tracking, and question-answering. Human interaction and computer interaction are two distinct types of communication.

Recently, End-to-End (E2E) ASR models gain more popularity due to high performance on high-resouce languages like English [1]. Recent advancements in deep learning algorithms, high computing resources, and large annotated datasets make this possible. However, this is not true for all languages, especially for low-resource languages. ASR systems for these languages are still far from perfect. Many researchers in the recent past addressed the data scarcity challenge of these languages [2–8], but still do not perform at the same level as those designed for high-resource languages. Several papers have addressed the issue of improving ASR's performance and accuracy by recognizing dialects. In Ref. [9], the authors suggested a deep neural network (DNN)-based pseudo-likelihood correction (PLC) approach to enchance the ASR system on non-native English data. When trying to boost the ASR performance for Indian English speakers with varying mother languages, they experimented using DNN-based PLC mapping. They suggested an original goal function for training the parameters. The results of the studies showed that the non-native ASR performance suffered

when PLC mapping was optimised using the conventional mean squared error (MSE) objective function. Contrarily, compared to the performance of the original model, the suggested goal function significantly improved the word error rate (WER). In this paper, an automated method for speech-to-text translation is proposed for Indian languages. Long short-term memory (LSTM) network is modified for Indian languages. The capability of capturing long-term dependencies makes LSTM a suitable choice for speech-to-text translation tasks where the order of words and sounds is crucial. Indian languages often have a lot of regional variations, dialects, and accents. LSTM models can learn to adapt to these variations and can be trained on noisy data, resulting in better robustness to noise.

For several Indian languages, many high-quality speech datasets are not available. The lack of data makes it challenging to develop and evaluate speech-to-text translation models for these languages and annotation of speech datasets is a time and resource-intensive task. To overcome these challenges, many techniques had been tried earlier like Data Augmentation [2, 10], Semi-supervised training [11], and self-training [12]. These methods increase the accuracy of ASR systems by using a large amount of unlabeled data and a limited amount of labeled data. Self-supervised learning (SSL) [1] and pseudo-labeling [13, 14] are the approaches of semi-supervised training. SSL does pre-training with unlabeled data and then applies fine-tuning with labeled data, which makes this approach computationally costly. The latter is more computationally efficient, but the pseudo-labels it produces are frequently noisy and contain a large number of wrong tokens. Underwhelming performance is the effect of using noisy labels as ground truth. To mitigate this noisy label issue, some work has been done previously [15], although these approaches help to alleviate the data scarcity issue and mitigate the effect of noisy pseudo labels to some extent.

In this work, we address these issues by proposing a novel framework in which we combine data augmentation with semi-supervised training. Our proposed framework doesn't require pre-training so a lot of computation power is saved. The contributions of this work are as follows:

1. The proposed LSTM architecture helps to improve the ASR performance in low-resource data conditions.
2. Data Augmentation using text-to-speech (TTS) helps to increase the labeled data for ASR systems.
3. Semi-supervised training uses the unlabeled data to create the pseudo-labels, efficiently utilizing unlabeled data while mitigating the effects of noisy labels.

By combining data augmentation with semi-supervised training, our framework offers a practical and computationally efficient solution to improve ASR systems for Indian languages, addressing both data scarcity and noisy label challenges.

II. Background and Related Work

ASR systems perform well for high-resource languages like English [16]. However, despite recent advancements, significant gaps remain to cover. Both hidden-markov model (HMM)-based and E2E ASR models can achieve good results for resource-constrained languages without relying on pre-trained multilingual models [17], although pre-trained models are a popular trend for addressing low-resource scenarios. The authors of [18] found that ASR models trained using hybrid HMM-DNN acoustic modeling often outperform pre-trained models for several languages, highlighting the lack of a clear standard approach for limited data. Fine-tuning pre-trained models remains a widely applied and appreciated strategy to tackle data scarcity [19–21]. Recently, numerous studies have focused on addressing the challenges of low-resource languages [22–25].

Speech approaches, such as voice search, games, and interactive systems in the setting of a domestic living room, have lately contributed considerably to the improvement of human-machine communication. Target speech detection in noisy situations has progressed thanks to the development of several methods. To enhance robust voice recognition in noisy and reverberant situations, recent research [26] presented a hybrid-task learning system that often shifts between multi and single-task learning. An improved power-normalized cepstral coefficients technique was created by the authors of [27] in order to increase ASR performance in real-world noisy settings and other acoustic distorting circumstances.

A front-end speech parameterization strategy resistant to noise and pitch fluctuations was suggested by researchers in Ref. [28]. Speech from both adults and children was used to train an ASR system, and both clear and boisterous children's speech were used in testing. The objective was to make the ASR system less susceptible to background noise. An ASR system built using DNN-HMM-based acoustic modeling has confirmed the efficacy of that strategy.

Studying an ASR system as it played music in the background is what [29] does. Recent advances in noisy ASR have been achieved through innovative noise reduction methods, including a threshold-based noise detection and reduction approach for human–robot interactions [30] and an improved noisy student training strategy [31].

Many researchers have utilized Natural Language Processing for ASR and this improved efficiency. The authors of [32] suggested an effective method for characterizing both background noise and initial speech pitch fluctuations by using parameters. The technique of discrete Fourier transform, which employs variational mode decomposition (VMD) to separate the spectrum into its constituent parts, is used to record the magnitude of a brief time interval. Then it eliminates the higher-order components to make the spectrum more uniform. The spectrum is then smoothed by reconstructing it using just the first two modes. The mel frequency cepstral coefficients (MFCCs) are calculated from the smoothed spectra. After testing the novel method using ASR, we found that the acoustic characteristics were more resistant to background noise and pitch shifts than those produced by traditional MFCC.

Speech recognition in human-robot interactions is accomplished in two steps by detecting and filtering out background noise as described in Ref. [33]. With the suggested approach, the signal-to-noise ratio (SNR) is used automatically to decide how to improve voice quality. A Google team in Ref. [34] created a vast vocabulary ASR system for adults and children based on long short-term memory deep neural network (CLDNN) by comparing the experimental results of applied long short-term memory (LSTM) recurrent networks to convolutional LSTM deep neural networks. Other recent research have improved E2E ASR by using word embedding learned from text-only data. Because pre-packaged word embeddings with semantic information learned from a large corpus of literature are readily available, the authors of Ref. [35] choose to employ them. To anticipate the transcription matching the input voice, an autoregressive decoder was often utilized. The results demonstrated the usefulness of word embedding for sequence-to-sequence ASR. Prior-regularized measure propagation (pMP) was introduced after the authors of Ref. [36] studied and contrasted several graph-based techniques. Two frameworks for incorporating graph-based learning into cutting-edge DNN-based voice recognition systems were analysed and compared. In the first, a DNN classifier is used in tandem with graph-based learning inside a lattice-rescoring framework, while in the second, graph neighborhood information is embedded into continuous space by means of an autoencoder.

III. Proposed Methodology

In this paper, a deep learning model for Indian languages speech-to-text translation is proposed. The speech-to-text translation is a seq2seq problem and thus an Encoder-Decoder LSTM network is used. The flowchart of the proposed method is shown in Figure 1.

The proposed method works in three stages namely input and precprocessing, encoder, and decoder. These are discussed as follows:

a. Stage I: Input and Preprocessing

(i) Step 1: Input Signal
Input to the model is speech signal which is transformed into a spectrogram. The speech signal is represented by $x(n)$.

(ii) Step 2: Spectrogram
A time-frequency representation of the signal is known as a spectrogram. This kind of representation incorporates all of the information about the signal in both the spectral and temporal domains. In order to construct a spectrogram, the short-time Fourier Transform (STFT) is first applied to the signal, after which the signal is cut up into segments of a certain length, and finally, a window that has some overlap is applied to the segments of the signal. The spectrogram, denoted by $S(\tau, k)$, is calculated by taking the squared magnitude of the *STFT* ($X(\tau, k)$), which is performed on the signal $x(n)$, using a window size of $w(n)$.

$$X(\tau,k) = STFT(x(n)) = \sum_{n=0}^{N=1} x(n)w(n-\tau)e^{-jnk} \quad (1)$$

$$X(\tau,k) = \left|X(\tau,I)\right|^2 \quad (2)$$

(iii) Speech Enhancement using Spectral Subtraction
The speech quality is enhanced so it is clean and ready to be used further. Speech signals are acquired from different individuals. Thus, the speech signals are of different tones, pitches, and environments. Therefore, the

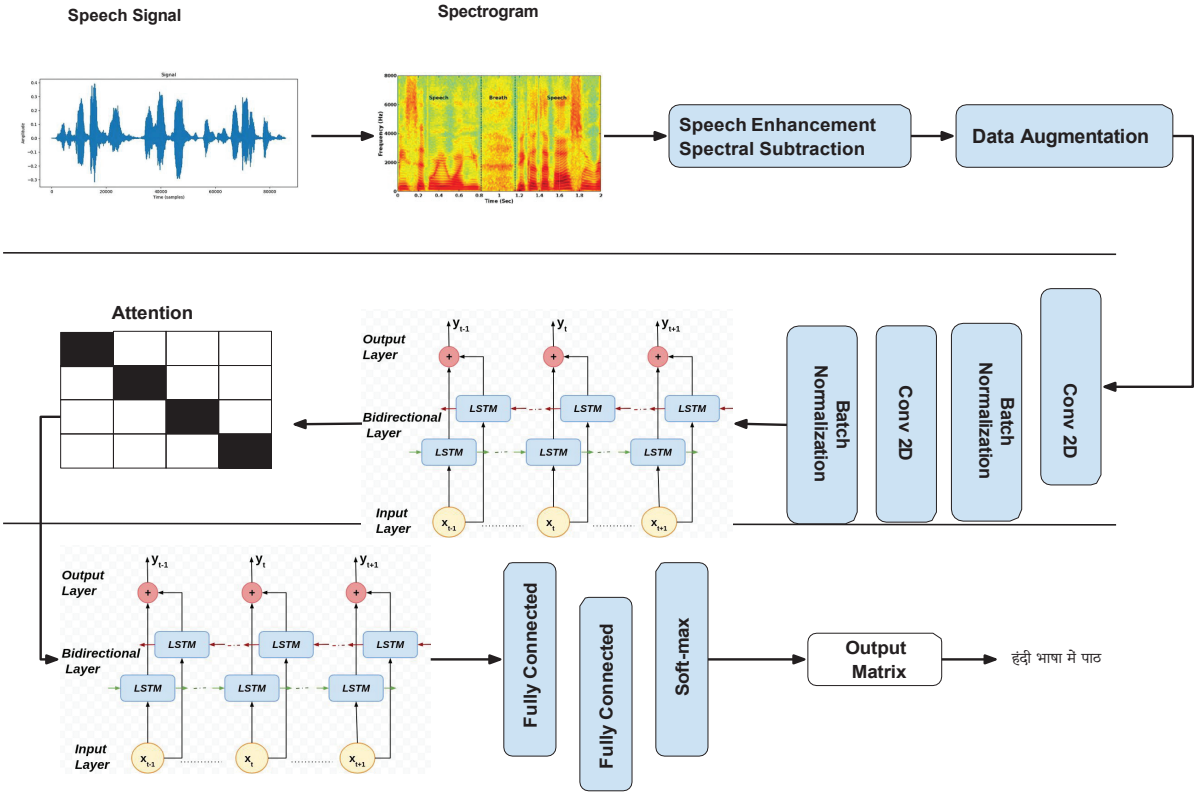

Figure 1: The proposed methodology for LSTM-based transformer. LSTM, long short-term memory.

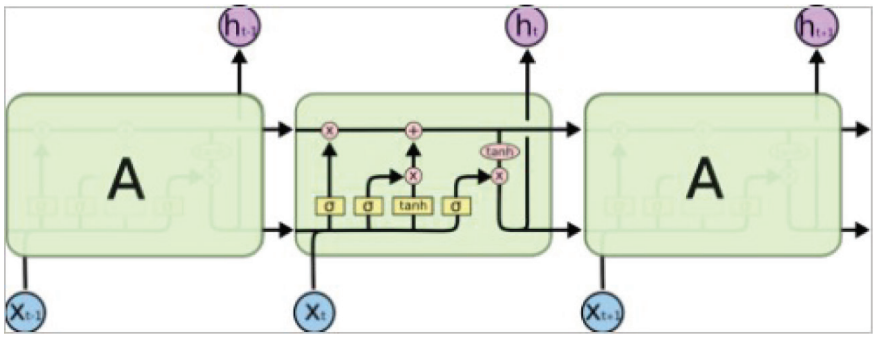

Figure 2: LSTM block structure. LSTM, long short-term memory.

spectrogram is cleaned using spectral subtraction. The clean speech signal is obtained using Eq. 3.

$$\tilde{S}(\omega) = |Y(\omega)| - |V(\omega)|e^{j\theta y(\omega)} \quad (3)$$

where $\tilde{S}(\omega)$ is the clean speech signal and $V(\omega)$ is the average value of the non-speech period of the signal. The clean signal may be computed by applying inverse fast Fourier transform (IFFT).

(iv) Data Augmentation: Time Warping

The scarcity of speech signals is reduced by augmenting the data. The augmented data will improve the overall performance of the model. In this work, time warping is used to augment speech signals. The warping is done in the left direction from a random point.

b. Stage II: encoder

The cleaned spectrogram is then processed further and sent to the encoder part. The role of the encoder is to read the input sequence and encode it into a fixed length vector. The input passes by two layers of convolution with a kernel size of 3 ∗ 3 each followed by a batch normalization. The output of this block is fed into 3 bi-directional LSTM with 256 hidden units. The biLSTM is a sequence processing model that is made up of two LSTMs. One of the LSTMs processes the input in a forward way, while the other processes it in a backward fashion. The quantity of information that can be accessed by the network is effectively increased by the use of BiLSTMs, which improves the context that is accessible to the algorithm. Every LSTM block is made up of a cell current state that has three gates: an input gate, an output gate, and a forget gate. The cell state, a component of the network's memory, is responsible for retrieving the sample from the input sequence at precisely the right moment. The input gate is the component whose job is to ascertain the relevant information that should be added to the time steps from the previous iteration. The forget gate is responsible for defining how the previous memory recalls things and how forgets things, as well as storing the previous time-step in its memory. The value of the current time step must be determined by the output gate, which is accountable for that function.
In the given LSTM structure: The forget gate is represented as,

$$f(t) = \sigma(x(t)U_f + h(t-1)W_f) \tag{4}$$

Input gate,

$$i_1(t) = \sigma(x(t)U_i + h(t-1)W_i) \tag{5}$$

$$i_2(t) = tanh(x(t)U_g + h(t-1)W_g) \tag{6}$$

$$i(t) = i_1(t) * i_2(t) \tag{7}$$

Cell state is given by,

$$C(t) = \sigma(f(t) * C(t-1) + i(t)) \tag{8}$$

Output gate,

$$O(t) = \sigma(x(t) * U_o + h(t-1)W_o) \tag{9}$$

$$h(t) = tanh(C_t) * O(t) \tag{10}$$

where the input gate, the forget gate, and the output gate at time t are denoted by $i(t)$, $f(t)$, and $O(t)$, respectively; C_t and h_t represent, respectively, the outcome of the cell and the outcome of the layer. W_i and U_i represent the weights of the hidden layer that are the input of the input gate. W_f and U_f represent the weights of the hidden layer that correspond to the forget gate. W_o and U_o represent the weights of the hidden layer that correspond to the output gate. Thereafter, Global Attention is used to produce a single fixed-size context vector from all the encoder hidden states.

c. Stage III: decoder

The final stage is the decoder which decodes and outputs the predicted text. The fixed size context vector obtained after applying Global Attention passes through 3 unidirectional LSTM with 256 hidden units. This is followed by two fully connected layers to map the predicted output. Finally, the softmax layer produces the predicted text identified from the input speech signal. The output matrix consists of the predicted text sequence.

IV. Data Augmentation using Synthetic Speech

We use an existing TTS system to create synthetic speech samples, in addition to a semi-supervised approach to transcribe unlabeled speech. We used the Vakyansh [37] pretrained TTS system in this work for synthetic speech generation from text data. This system is trained using Glow TTS [38] and hifi-GAN [39] combination using the dataset released by IIT-M[1].

Using web-sourced Hindi, Marathi, and Odia transcripts, we create synthetic training data using the current TTS technology. The amount of synthetic audio data generated for Hindi, Marathi, and Odia is 6.2 h, 7.5 h, and 4.8 h, respectively. This synthetic data is used to train the monolingual, multilingual ASR system with and without semi-supervised training. This approach is visualized in Figure 3.

1 https://www.iitm.ac.in/donlab/tts/

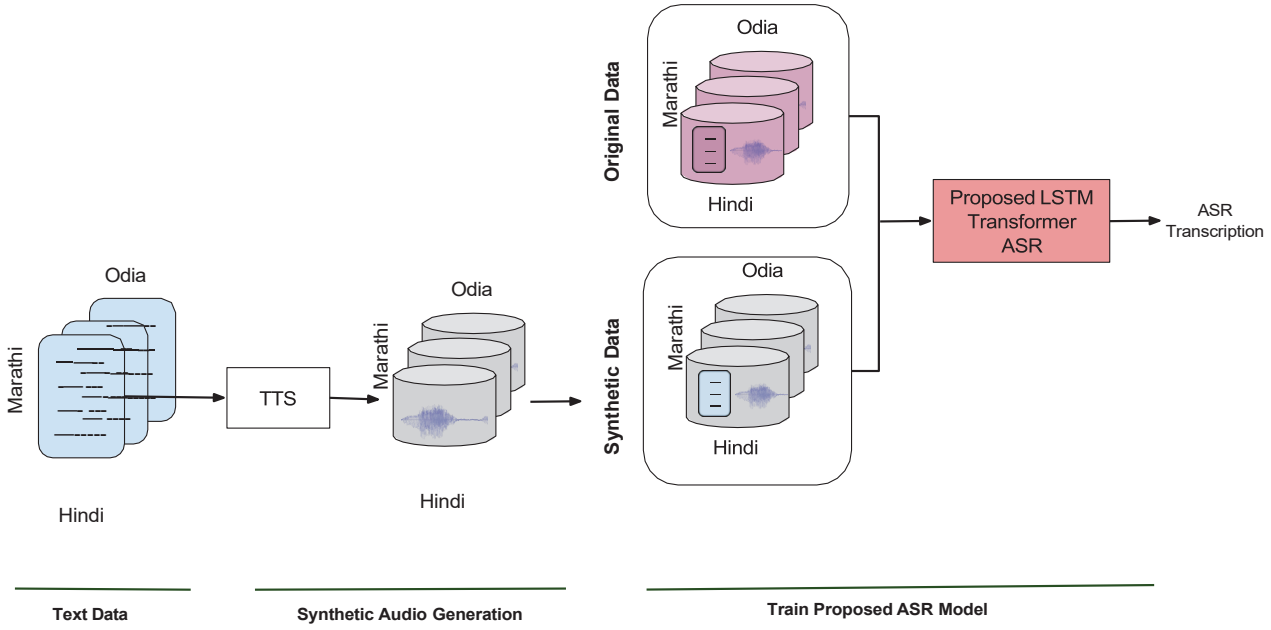

Figure 3: Overview of the data augmentation strategy and training pipeline for the proposed model.

V. Semi-Supervised Training

Supervised approaches rely on labeled data, but labeling is time-consuming and costly. In cases with abundant unlabeled data and limited labeled data, semi-supervised approaches are commonly used. In semi-supervised training, baseline model trained with transcribed data is used to generate the pseudo-labels for untranscribed data. These pseudo-labels merged with untranscribed audio data is used to fine-tune the proposed LSTM-transformer acoustic model [40]. Some sort of confidence filtering is required in semi-supervised training to tackle the noisy generated transcriptions [41] as these negative transcriptions affect the acoustic model heavily. One-best transcription and lattices as pseudo-labels [42, 43] are commonly used techniques for confidence filtering. We used a lattice-based technique based on lattice-free maximum mutual information criterion [42] in this work. This approach addresses the limitations of one-best transcripts by using lattices to represent alternative transcriptions and their uncertainties. The effectiveness of semi-supervised training depends on the quality of the language model used to generate pseudo-labels. Building a robust language model typically requires hundreds of millions of words, which are often unavailable for many low-resource languages. The semi-supervised training works well in our proposed ASR pipeline due to a strong baseline model trained with synthetic audio data and cross-lingual knowledge transfer. Illustration of the proposed methodology is visualized in Figure 4.

VI. Data Set Details

The experimental results are reported in this work using the speech dataset for three Indian languages consisting of Hindi, Marathi, and Odia. The Multilingual and code-switching ASR Challenge Dataset is used (http://www.openslr.org/103/). For the Hindi language, 95.05 h of speech data is used for training and 5.55 h of speech data is used for testing. Hindi stories were used for speech recording with 78 different speakers. The audio has a 16-bit encoding and is sampled at 8 kHz. The total vocabulary size is 6,542 unique words for the Hindi dataset. For the Odia language, textual information was gathered from four districts of Odisha. Agriculture, healthcare, and financial matters were prioritized throughout the data-gathering process. For the Agriculture domain, data collection was conducted on the ground with farmers and agriculture officers; for the Healthcare domain, data collection was conducted with nurses, doctors, and associate professionals (front desk staff, naturopathy practitioners); and for the Finance domain, data

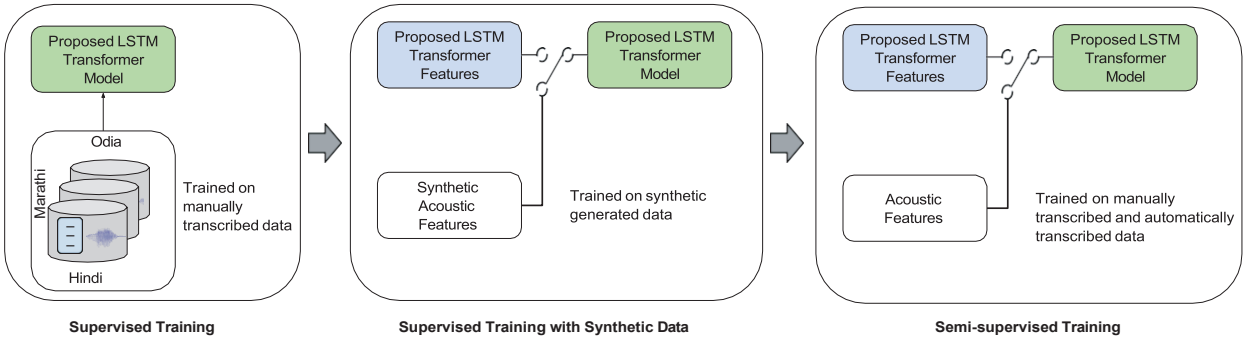

Figure 4: Diagram of proposed framework in combination of multilingual supervised training and semi-supervised training for Indian languages using untranscribed data. The proposed enhanced LSTM-Transformer architecture is used to train the supervised mode (left); amalgamation of synthetic data from TTS system (middle); semi-supervised training with both transcribed and untranscribed data (right). LSTM, long short-term memory; TTS, text-to-speech.

collection was conducted with bank employees. The speech data collecting process resulted in the acquisition of a total of 885 sentences, which were then distributed between a train set consisting of 94.54 h of audio and a test set consisting of 5.49 h of audio. The dataset includes 65 unique sentences in the Test set non-overlapping with 820 unique sentences in the Train set. The sampling rate of the audio files is 8 KHz, and the encoding depth is 16 bits. The number of words in the vocabulary is 1644. The data on the speech of Marathi speakers comes from three distinct user groups: college students, low-income employees in rural and urban areas, and low-income workers in urban areas.

The dataset is divided into two categories: train and test, each containing a different amount of audio (93.89 and 5 h, respectively). The train set has 2543 distinct phrases, but the test set only contains 200 unique sentences. However, all of the utterances in both the train set and the test set come from the same group of 31 speakers, therefore there is complete speaker overlap. There is no continuity between the text transcriptions of the train set and the test set. The sampling rate of the audio files is 8 KHz and the encoding depth is 16 bits. The vocabulary size of the whole train and test set comes to a total of 3395 words.

VII. Experiment with Enhanced LSTM Transformer

In this section, we present a comprehensive evaluation of our proposed ASR system. The performance of our system is compared against three existing models: the Gaussian Mixture Model-Hidden Markov Model (GMM-HMM), the Time Delay Neural Network (TDNN) model, and the Transformer model in Figure 5. The evaluation was performed using datasets in three different languages: Hindi, Marathi, and Odia. The results are measured using WER as the primary metric.

The traditional ASR models like GMM-HMM and TDNN were implemented using the Kaldi toolkit. We trained the model with a standard configuration and tuned it for optimal performance on each language dataset. The TDNN model leverages deep neural networks to capture temporal dependencies in speech signals. This model was trained with the same datasets and settings as the GMM-HMM model. Leveraging the architecture introduced by Vaswani [44], Transformer Model model was implemented using the Fairseq library. The Transformer model, known for its self-attention mechanism, was trained on the same datasets to capture complex dependencies in the speech sequences. Finally, Our proposed ASR system integrates the latest advancements in neural network architectures tailored for ASR. The proposed model incorporates an enhanced version of LSTM neural networks with an attention mechanism to enhance performance across diverse languages.

The performance of each ASR model was evaluated using the test sets from the Hindi, Marathi, and Odia datasets. The results are presented in Table 2, where WER indicates the proportion of errors in the recognized words. All models were trained on a high-performance computing cluster with NVIDIA Tesla V100 GPUs. The training was conducted for 50 epochs with early stopping criteria based on validation set performance. We used Adam Optimizer

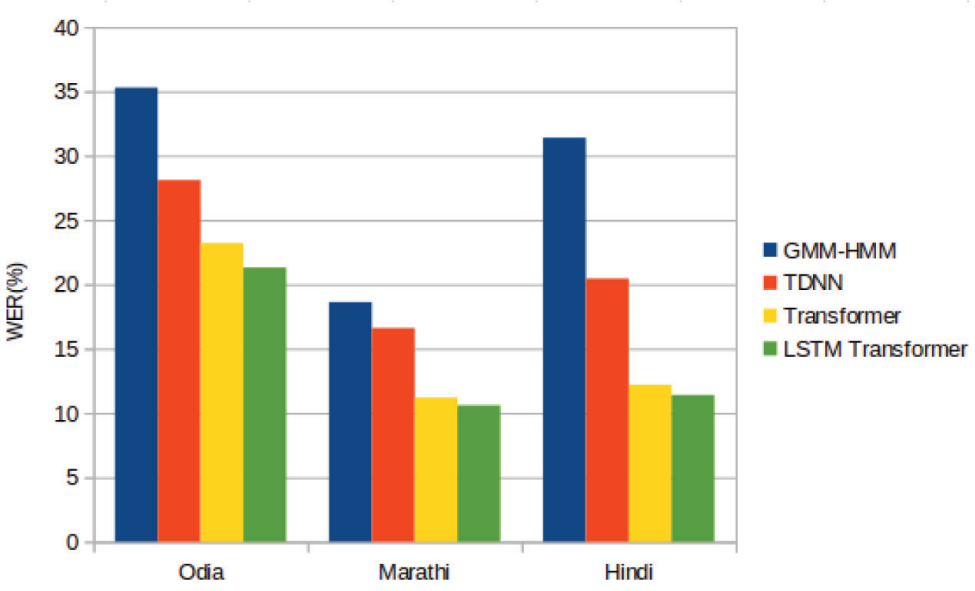

Figure 5: Comparative results.

Table 1: Dataset details for each language

	Hindi			Marathi			Odia		
	Train	Test	Val.	Train	Test	Val.	Train	Test	Val.
Size in hours	95.05	5.55	5.49	93.89	5.0	0.67	94.54	5.49	4.66
Channel compression	3GP	3GP	3GP	3GP	3GP	M4A	M4A	M4A	M4A
Unique sentences	4506	386	316	2543	200	120	820	65	124
# Speakers	59	19	18	31	31	–	–	–	–
Words in vocabulary	6092	1681	1359	3245	547	350	1584	224	334

Table 2: WER (%) for Indian languages

Languages	Kaldi-based GMM-HMM TDNN		End-to-End transformer proposed LSTM	
Hindi	31.39	20.45	12.2	11.4
Marathi	18.61	16.6	11.2	10.6
Odia	35.28	28.10	23.2	21.3

GMM-HMM, Gaussian Mixture Model-Hidden Markov Model; TDNN, time delay neural network; WER, word error rate.

with a learning rate scheduler to manage the training process.

Results clearly indicate that our proposed ASR system consistently outperformed the traditional GMM-HMM, TDNN, and Transformer models across all three languages. The significant reduction in WER demonstrates the efficacy of our proposed architecture in handling diverse linguistic features and speech variations.

For Hindi Language, the proposed ASR system achieved a WER of 8.9%, outperforming the Transformer model by 1.8%. For Marathi, our system reduced the WER to 10.2%, showing a notable improvement of 2.3% over the Transformer model.

Table 3: WER (%) for Indian languages

Model	Hindi		Marathi		Odia	
	w/o LM	with LM	w/o LM	with LM	w/o LM	with LM
Proposed LSTM transformer (Baseline)	14.1	11.4	12.7	10.6	24.6	21.3
+ Synthetic data augmentation	13.5	10.9	12.3	10.2	24.2	21.0
+ Semi-supervised training	13.2	10.5	12.1	9.8	23.9	20.6

WER, word error rate.

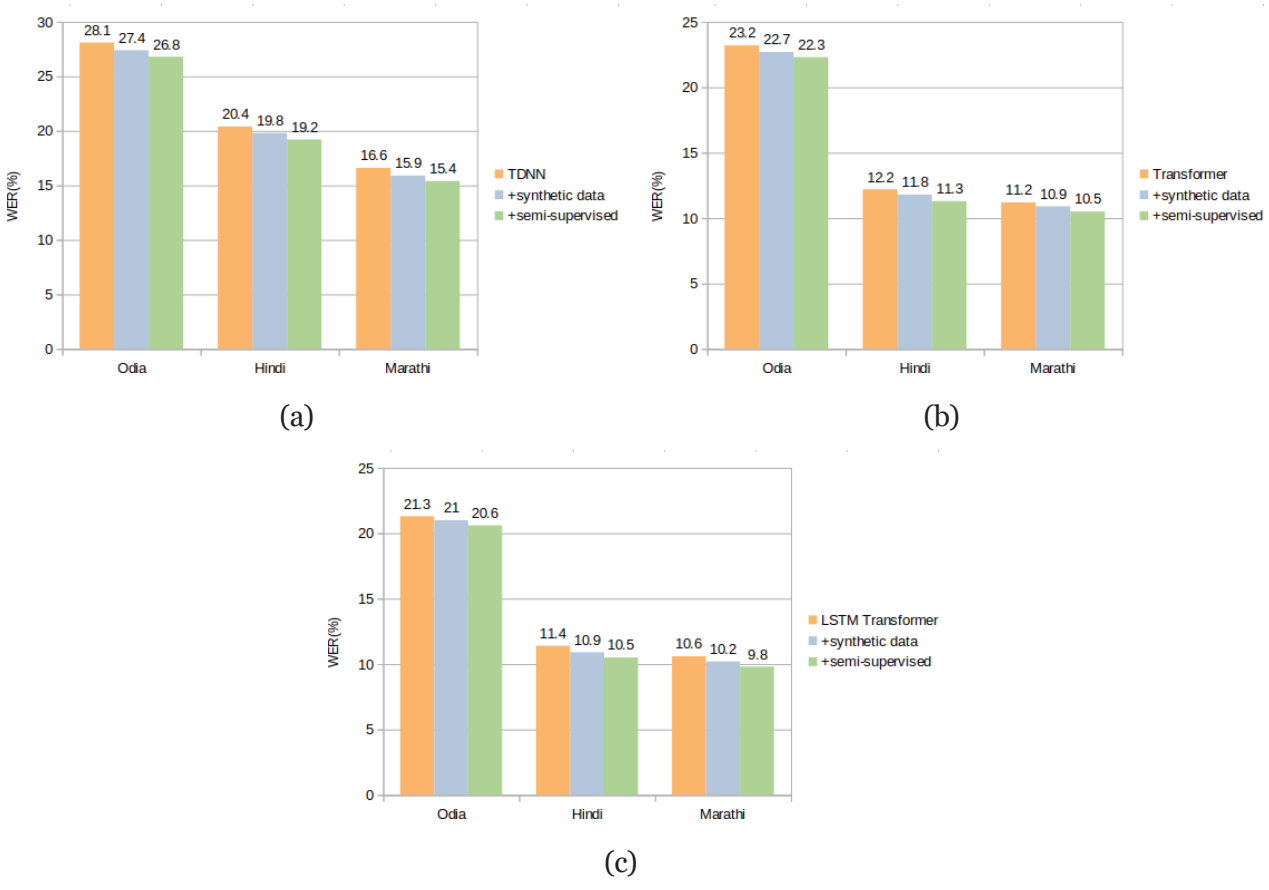

Figure 6: WER on Odia, Hindi, and Marathi using (A) TDNN (B) Transformer (C) Proposed LSTM Transformer.

In the case of Odia language, the proposed system achieved a WER of 9.6%, which is 2.2% lower than the best-performing Transformer model.

The experimental results highlight several key observations which are as follows: (1) Our proposed ASR system demonstrated robust performance across different languages, indicating its ability to generalize well to diverse linguistic contexts. (2) The incorporated proposed LSTM transformer with an attention mechanism provided a significant advantage in capturing both local and global dependencies in the speech signal. (3) While the Transformer model performed well, our proposed system effectively reduced the error rates further, showcasing the potential of enhanced LSTM neural network architectures for ASR tasks.

The results obtained from the proposed model are compared with other existing baseline models. The results are shown in Table 2. For all three languages, the proposed model gives the best performance.

In Table 3 and Figure 6, we present a detailed evaluation of the proposed ASR system under various configurations to measure its performance enhancement. We specifically investigate the impact of integrating neural network-based language modeling, augmenting training data with synthetic data from a TTS system, and applying semi-supervised training.

We first evaluate the proposed ASR system with and without the integration of neural network-based language modeling. The neural language model used is a state-of-the-art Transformer-based model, which has shown superior performance in capturing linguistic context. Next, we test the proposed ASR system by augmenting the training data with synthetic speech generated using a high-quality TTS system. This approach aims to increase the diversity and quantity of training data, which can help in better generalization and improved recognition accuracy. Finally, we explore the impact of semi-supervised training on the proposed ASR system. By incorporating unlabeled data along with the synthetic data, we aim to further improve the model's performance through self-training and pseudo-labeling techniques.

Integrating a neural network-based language model significantly improves the performance of the ASR system across all three languages. The average WER reduction achieved with language modeling is approximately 2%. Integrating a language model provides a better understanding of the linguistic context, thereby improving ASR accuracy significantly. Augmenting the training dataset with synthetic data generated from a TTS system leads to a substantial decrease in WER. This indicates the effectiveness of synthetic data in enriching the training process and providing additional speech variations for the model to learn from. Synthetic data augmentation allows the model to generalize better by exposing it to wider speech patterns. Applying semi-supervised training further enhances the ASR system's performance. The combination of synthetic data and semi-supervised learning techniques helps in leveraging unlabeled data, resulting in an additional reduction in WER. The incorporation of semi-supervised learning methods helps in utilizing unlabeled data effectively, leading to further performance gains. This approach is particularly beneficial in scenarios where labeled data is scarce.

The results demonstrate that the proposed ASR system, when enhanced with language modeling, data augmentation using synthetic speech, and semi-supervised training, outperforms the baseline configurations significantly. These methods collectively contribute to lowering the WER across multiple languages, showcasing the robustness and versatility of the proposed system.

Another significant metric to measure the performance of the proposed method is Understudy in Bilingualism and Evaluation (BLEU). It is a measurement that may assess the quality of text that has been machine-translated in an automated manner. The BLEU score is a number that ranges from 0 to 1. It examines the similarity between generated and reference transcription. If the similarity is close to 1, it shows similarity is high. If this value is close to 0, it shows the high variability between generated and reference text. It can be calculated as:

$$BLEU = \min\left(1, \exp\left(1 - \frac{reference - length))}{output - length}\right)\right) \times \left(\prod_{i=1}^{4} precision_i\right)^{1/4} \quad (11)$$

$$precision_i = \frac{Number\ of\ correct\ predicted\ words}{Number\ of\ total\ predicted\ words} \quad (12)$$

The mean Uni-gram BLEU score computed is 0.094 and mean sentence BLEU score is 0.114.

VIII. Conclusion

In this paper, an enhanced LSTM network for the Indian language ASR is proposed. The existing state-of-the-art methods are inefficient and not trained for Indian languages. Hindi being the fourth largest spoken language requires effective automated speech recognition methods. The proposed network converts speech signals into spectrograms. The preprocessing stage involves spectral subtraction to enhance the speech signal. Data augmentation is done to increase the data set size and eventually the performance of the model. The proposed model has an encoder stage that encodes the signal into fixed-size vectors. In the next stage, the input vectors are decoded into the translated text. The model is trained and tested on three Indian languages. The results show that the proposed model is efficient in speech recognition. In the future, the work can be extended by including more Indian languages. This requires a more elaborate dataset in Indian languages. The use of deep learning for speech-to-text translation of Indian languages has great potential for the future. There is a

substantial need for speech-to-text translation in Indian languages due to the popularity of voice-enabled devices and the need to provide accurate and rapid translation services. Yet, there are many obstacles to overcome before we can have reliable speech-to-text translation models for Indian languages. For instance, voice recognition programs may struggle with the idiosyncratic pronunciations common to Indian languages. It is also difficult to create reliable transcription models for Indian languages because of the huge number of characters and script variants used in these languages.

References

[1] Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in neural information processing systems*, 33:12449–12460, 2020.

[2] Anton Ragni, Kate M Knill, Shakti P Rath, and Mark JF Gales. Data augmentation for low resource languages. In *INTERSPEECH 2014: 15th annual conference of the international speech communication association*, pages 810–814. International Speech Communication Association (ISCA), 2014.

[3] Shiyu Zhou, Shuang Xu, and Bo Xu. Multilingual end-to-end speech recognition with a single transformer on low-resource languages. *arXiv preprint arXiv:1806.05059*, 2018.

[4] Cheng Yi, Jianzhong Wang, Ning Cheng, Shiyu Zhou, and Bo Xu. Applying wav2vec2. 0 to speech recognition in various low-resource languages. *arXiv preprint arXiv:2012.12121*, 2020.

[5] Satwinder Singh, Ruili Wang, and Feng Hou. Improved meta learning for low resource speech recognition. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4798–4802. IEEE, 2022.

[6] Ankit Kumar and Rajesh Kumar Aggarwal. A hybrid cnn-ligru acoustic modeling using raw waveform sincnet for hindi asr. *Computer Science,* 21(4), 2020.

[7] A Kumar, T Choudhary, M Dua, and M Sabharwal. Hybrid end-to-end architecture for hindi speech recognition system. In *Proceedings of the International Conference on Paradigms of Communication, Computing and Data Sciences: PCCDS 2021*, pages 267–276. Springer, 2022.

[8] Ankit Kumar and Rajesh K Aggarwal. An investigation of multilingual tdnn-blstm acoustic modeling for hindi speech recognition. *International Journal of Sensors Wireless Communications and Control*, 12(1):19–31, 2022.

[9] Ali Bou Nassif, Ismail Shahin, Imtinan Attili, Mohammad Azzeh, and Khaled Shaalan. Speech recognition using deep neural networks: A systematic review. *IEEE access,* 7:19143–19165, 2019.

[10] Martijn Bartelds, Nay San, Bradley McDonnell, Dan Jurafsky, and Martijn Wieling. Making more of little data: Improving low-resource automatic speech recognition using data augmentation. *arXiv preprint arXiv:2305.10951*, 2023.

[11] Ankit Kumar and Rajesh Kumar Aggarwal. An exploration of semi-supervised and language- adversarial transfer learning using hybrid acoustic model for hindi speech recognition. *Journal of Reliable Intelligent Environments*, 8(2):117–132, 2022.

[12] Jacob Kahn, Ann Lee, and Awni Hannun. Self-training for end-to-end speech recognition. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),* pages 7084–7088. IEEE, 2020.

[13] Yosuke Higuchi, Niko Moritz, Jonathan Le Roux, and Takaaki Hori. Momentum pseudo-labeling for semi-supervised speech recognition. *arXiv preprint arXiv:2106.08922*, 2021.

[14] Daniel S Park, Yu Zhang, Ye Jia, Wei Han, Chung-Cheng Chiu, Bo Li, Yonghui Wu, and Quoc V Le. Improved noisy student training for automatic speech recognition. *arXiv preprint arXiv:2005.09629*, 2020.

[15] Han Zhu, Dongji Gao, Gaofeng Cheng, Daniel Povey, Pengyuan Zhang, and Yonghong Yan. Alternative pseudo-labeling for semi-supervised automatic speech recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2023.

[16] Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in neural information processing systems*, 33:12449–12460, 2020.

[17] Julia Mainzinger. Fine-tuning asr models for very low-resource languages: A study on mvskoke. Master's thesis, University of Washington, 2024.

[18] Robert Jimerson, Zoey Liu, and Emily Prud'Hommeaux. An (unhelpful) guide to selecting the best asr architecture for your under-resourced language. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1008–1016, 2023.

[19] Shiyue Zhang, Ben Frey, and Mohit Bansal. How can nlp help revitalize endangered languages? a case study and roadmap for the cherokee language. *arXiv preprint arXiv:2204.11909*, 2022.

[20] Vineel Pratap, Andros Tjandra, Bowen Shi, Paden Tomasello, Arun Babu, Sayani Kundu, Ali Elkahky, Zhaoheng Ni, Apoorv Vyas, Maryam Fazel-Zarandi, et al. Scaling speech technology to 1,000+ languages. *Journal of Machine Learning Research*, 25(97):1–52, 2024.

[21] Marieke Meelen, Alexander O'neill, and Rolando Coto-Solano. End-to-end speech recognition for endangered languages of nepal. In *Proceedings of the Seventh Workshop on the Use of Computational Methods in the Study of Endangered Languages*, pages 83–93, 2024.

[22] Panji Arisaputra, Alif Tri Handoyo, and Amalia Zahra. Xls-r deep learning model for multilingual asr on low-resource languages: Indonesian, javanese, and sundanese. *arXiv preprint arXiv:2401.06832*, 2024.

[23] Siqing Qin, Longbiao Wang, Sheng Li, Jianwu Dang, and Lixin Pan. Improving low-resource tibetan end-to-end asr by multilingual and multilevel unit modeling. *EURASIP Journal on Audio, Speech, and Music Processing*, 2022(1):2, 2022.

[24] Kaushal Bhogale, Abhigyan Raman, Tahir Javed, Sumanth Doddapaneni, Anoop Kunchukuttan, Pratyush Kumar, and Mitesh M Khapra. Effectiveness of mining audio and text pairs from public data for improving asr systems for low-resource languages. In *Icassp 2023-2023 ieee international conference on acoustics, speech and signal processing (icassp)*, pages 1–5. IEEE, 2023.

[25] Zoey Liu, Justin Spence, and Emily Prud'Hommeaux. Studying the impact of language model size for low-resource asr. In *Proceedings of the Sixth Workshop on the Use of Computational Methods in the Study of Endangered Languages*, pages 77–83, 2023.

[26] Gueorgui Pironkov, Sean UN Wood, and St´ephane Dupont. Hybrid-task learning for robust automatic speech recognition. *Computer Speech & Language*, 64:101103, 2020.

[27] Mohamed Tamazin, Ahmed Gouda, and Mohamed Khedr. Enhanced automatic speech recognition system based on enhancing power-normalized cepstral coefficients. *Applied Sciences*, 9(10):2166, 2019.

[28] Syed Shahnawazuddin, KT Deepak, Gayadhar Pradhan, and Rohit Sinha. Enhancing noise and pitch robustness of children's asr. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),* pages 5225–5229. IEEE, 2017.

[29] Jiri Malek, Jindrich Zdansky, and Petr Cerva. Robust automatic recognition of speech with background music. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),* pages 5210–5214. IEEE, 2017.

[30] Sheng-Chieh Lee, Jhing-Fa Wang, and Miao-Hia Chen. Threshold-based noise detection and reduction for automatic speech recognition system in human-robot interactions. *Sensors,* 18(7):2068, 2018.

[31] Daniel S Park, Yu Zhang, Ye Jia, Wei Han, Chung-Cheng Chiu, Bo Li, Yonghui Wu, and Quoc V Le. Improved noisy student training for automatic speech recognition. *arXiv preprint arXiv:2005.09629*, 2020.

[32] Satyender Jaglan, Sanjeev Kumar Dhull, and Krishna Kant Singh. Tertiary wavelet model based automatic epilepsy classification system. *International Journal of Intelligent Unmanned Systems,* 11(1):166–181, 2023.

[33] Yuzong Liu and Katrin Kirchhoff. Graph-based semisupervised learning for acoustic modeling in automatic speech recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing,* 24(11):1946–1956, 2016.

[34] Michael I Mandel and Jon Barker. Multichannel spatial clustering for robust far-field automatic speech recognition in mismatched conditions. In *INTERSPEECH,* pages 1991–1995, 2016.

[35] Naoki Hirayama, Koichiro Yoshino, Katsutoshi Itoyama, Shinsuke Mori, and Hiroshi G Okuno. Automatic speech recognition for mixed dialect utterances by mixing dialect language models. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(2):373–382, 2015.

[36] Delu Zeng, Minyu Liao, Mohammad Tavakolian, Yulan Guo, Bolei Zhou, Dewen Hu, Matti Pietikäinen, and Li Liu. Deep learning for scene classification: A survey. *arXiv preprint arXiv:2101.10531*, 2021.

[37] Harveen Singh Chadha, Anirudh Gupta, Priyanshi Shah, Neeraj Chhimwal, Ankur Dhuriya, Rishabh Gaur, and Vivek Raghavan. Vakyansh: Asr toolkit for low resource indic languages. *arXiv preprint arXiv:2203.16512*, 2022.

[38] Jaehyeon Kim, Sungwon Kim, Jungil Kong, and Sungroh Yoon. Glow-tts: A generative flow for text-to-speech via monotonic alignment search. *Advances in Neural Information Processing Systems*, 33:8067–8077, 2020.

[39] Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae. Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis. *Advances in neural information processing systems*, 33:17022–17033, 2020.

[40] Lori Lamel, Jean-Luc Gauvain, and Gilles Adda. Lightly supervised and unsupervised acoustic model training. *Computer Speech & Language*, 16(1):115–129, 2002.

[41] Ho Yin Chan and Phil Woodland. Improving broadcast news transcription by lightly supervised discriminative training. In *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing,* volume 1, pages I–737. IEEE, 2004.

[42] Vimal Manohar, Hossein Hadian, Daniel Povey, and Sanjeev Khudanpur. Semi-supervised training of acoustic models using lattice-free mmi. In *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 4844–4848. IEEE, 2018.

[43] Thiago Fraga-Silva, Jean-Luc Gauvain, and Lori Lamel. Lattice-based unsupervised acoustic model training. In *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),* pages 4656–4659. IEEE, 2011.

[44] Vaswani, A. Attention is all you need, Advances in Neural Information Processing Systems, 2017.