Journal of Information Systems Engineering and Management
2025, 10(58s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Dialect Information based Manipuri Automatic Speech Recognition System in a Multi-task framework using well-trained Self Supervised Learning Based models

Thangjam Clarinda Devi[1*], Kabita Thaoroijam[2], Kishorjit Nongmeikapam[1]
[1]*Department of Computer Science and Engineering, Indian Institute of Information Technology Manipur, India*
[2]*School of Computer Science and Artificial Intelligence, SR University Warangal, India*
Corresponding author: thangc@iiitmanipur.ac.in

ARTICLE INFO

Received: 30 Dec 2024

Revised: 12 Feb 2025

Accepted: 26 Feb 2025

ABSTRACT

The process of transforming spoken language into text is known as Automatic Speech Recognition (ASR). Building ASR systems for low resourced languages is one of the greatest challenges within the field. One of the primary factors is the lack of adequately large corpora of transcribed speech datasets for these languages. This limitation may cause an inaccurate text transcription of the processed speech signals. Manipuri is low resource language selected as a case study owing to its rich unique socio-linguistic and cultural profiles. This paper analyzes the issues and progress made towards ASR system for spoken Manipuri language especially its dialect and the computing methods employed. Resources for self supervised feature extraction were applied to create Dialect Identification and ASR systems for Manipuri. ASR results improvement is observed with inclusion of dialect information during model training. The best results were attained with systems using Byte Pair Encoding (BPE) where a 19.8% Word Error Rate (WER) and 6.6% Character Error Rate (CER) were observed. This result is a benchmark and state-of-the-art for this dataset.

Keywords: *Automatic speech recognition, Dialect identification, Manipuri, Self supervised learning, Conformer*

INTRODUCTION

Automatic Speech Recognition (ASR) systems for low-resource languages (LRLs) are hindered by the unavailability of large-scale annotated datasets, limited linguistic resources, and complex dialectal variations. Many languages, particularly those spoken in regions with small populations or limited technological infrastructure, face challenges in developing robust ASR systems [1]. Dialectal variation is one of the most significant obstacles in the development of ASR models for LRLs, as it results in diverse phonetic realizations and can drastically affect speech recognition performance [2]. Recent advances in self-supervised learning (SSL) have introduced a new paradigm in speech processing, particularly for low-resource languages. Unlike traditional supervised learning, SSL allows the extraction of useful representations from unannotated speech data, making it ideal for languages with limited labeled resources [3]. SSL-based models can be pretrained on large corpora of unlabeled speech data and fine-tuned for specific dialects or recognition tasks. This paper reviews the use of SSL for speech recognition in low-resource languages, with an emphasis on dialectal variations.

Manipuri (Meiteilon), the official language of the state of Manipur, India, belongs to the Sino-Tibetan language family. It is predominantly spoken by the Meitei ethnic group, and its dialects exhibit significant phonological and lexical variations. The manipulation of phonetic structures, tone patterns, and prosody in speech poses a unique challenge in developing reliable speech recognition systems. With rapid advancements in machine learning and artificial intelligence, there is growing interest in adapting these technologies for languages like Manipuri, but this area remains underexplored. This paper aims to assess the state-of-the-art in dialect-based speech recognition for Manipuri, highlighting both progress and barriers in building robust systems for the language. The potential of SSL models is

investigated to overcome data scarcity, improve generalization across dialects, and facilitate the development of speech technologies for Manipuri language. The insights from this study can also be extended for other LRLs.

Our contributions in this study are summarized as follows:

1. Both dialect identification and low resource ASR have been developed for Manipuri language. The model achieves the state-of-the-art dialect identification and ASR results on this dataset. To the best of our knowledge, no previous works on the related tasks exist for this dataset.
2. The performance of different SSL features has been investigated for Transformer based and Conformer based ASR and dialect identification tasks.
3. The inclusion of dialect information during the ASR training process improves the ASR results, which have been demonstrated through empirical analysis.

This paper is organized as follows. Section 2 describes the related work. Section 3 explains the methodology and its implementation. Then, the experimental results are presented in Section 4. Discussion about the results obtained are shown in Section 5. Finally, Section 6 concludes our work.

RELATED WORK

In the last few years, there have been various types of approach to improve the performance of ASR systems for LRLs [4]. Most of these approaches consist of different data augmentation techniques to increase the size of usable training data. Speech synthesis based data augmentations as introduced by [5], where ASR models are finetuned using synthetic speech and text data and have shown significant improvement in the results of low resource ASR. Generative Adversarial Network (GAN) based augmentation [6] has also been explored to increase the amount of labeled data with the speech from LRLs. Some feature domain augmentation approach include Vocal Tract Length Perturbation [7]. Also audio based augmentations such as Speed Perturbation and Tempo Perturbation have been used. Two stage data augmentation on audio and feature-level seems to improve the ASR results for LRLs [4]. Spectrogram Augmentation also seems promising for improving the performance of ASR systems [8]. Each of these methods shows improvements in low resource ASR accuracy, however, they still require corresponding labeled annotations to speech data. Another recent trend is the use of transfer learning approaches for improving the recognition in low resource ASR for features adaptability from high resource to low resource speech. [9] have performed extensive analysis to understand the effect of the different values of the number of frozen layers in Deep Neural Network (DNN) transfer learning configurations, and their impact on improving low resource ASR. [6] have explored the use of a multi-stage training strategy, which involves the transfer learning approach using DeepSpeech and Gated Convolutional Neural Network models for improving the performance of ASR systems on low resource speech data. [10] have shown the performance of different neural network architectures for low resource ASR.

Some works on dialectal based ASR using SSL are mentioned below. The paper [11] focuses on accent identification and accented speech recognition using SSL. It proposes an accent-dependent ASR system that utilizes accent features, achieving a 6.5% relative word error rate reduction compared to an accent-independent ASR system. In [12], the authors proposes an end-to-end, transformers-based framework for Jordanian Arabic dialect speech-to-text using SSL. It incorporates noisy student training and data augmentation, significantly improving performance and enabling efficient processing of LRL data. The paper [13] explores dialect-based ASR using a SSL model adapted to a nationwide corpus of Japanese dialects. It highlights the model's effectiveness in recognizing various dialects and integrating dialect region identification tasks. A self-supervised multilingual speech model trained on African speech, achieving competitive results in ASR for dialects in sub-Saharan Africa, using significantly less data and parameters compared to existing models in [14]. The paper benchmarks state-of-the-art SSL speech encoders for ASR in the low-resource Tunisian Arabic dialect, evaluating their effectiveness in transcribing spoken utterances and understanding semantic content with limited training data [15]. In [16], SSL for ASR across 24 Indian languages, emphasizing multilingual pretraining has been explored. It demonstrates significant improvements in ASR performance, highlighting the effectiveness of SSL in handling diverse dialects within these languages.

Manipri ASR: [17] discussed a phonetic engine (PE) developed for Manipuri language speech recognition, utilizing hidden Markov models (HMM) for phonetic unit modeling. It compares three spectral features, finding that Perceptual Linear Prediction and Mel Frequency Cepstral Coefficients outperform Linear Prediction Cepstral Coefficients in recognition accuracy. The research [18] develops an ASR system for Manipuri, achieving a 13.57% Word Error Rate (WER) using DNN-HMM architecture, utilizing over 90 hours of telephonic read speech data from 300+ speakers. The research [19] introduces a bilingual Manipuri-English speech corpus for ASR, featuring 57 hours

Journal of Information Systems Engineering and Management
2025, 10(58s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

of annotated spontaneous speech. Various ASR models were developed, with the pure Time Delay Neural Network model demonstrating superior performance in recognizing Manipuri-English code-switched speech. The paper [20] focuses on Manipuri speech recognition by analyzing log mel features and wav2vec2 to build ASR in End to End fashion achieving 29.7% WER and 8.3% CER. As no dialect based ASR were found for Manipuri language, which uses SSL representation, the same has been implemented in order to bridge the gap.

METHODS

ULCA Manipuri dataset [21] is used for the experiment. The corpus was collected from All India Radio (AIR) Manipur led by EkStep Foundation under National Language Translation Mission (NLTM). The dataset consists of recordings of sentences from News domain recorded by male and female News readers. Dataset is about 10 hours of speech utterances. It has 5449 number of utterances in total. Also Linguistic Data Consortium for Indian Languages (LDCIL) [22] Manipuri speech corpus has been used which contains 10979 simple sentence utterances. The words used in these sentences are common words which are found in day-to-day life. Age of speakers varies from 16 to 51 years. All the speakers were literate enough and could read the textual sentences provided to them. Table 1 shows the dataset details. Table 2 shows the dialectal utterances count in ULCA+LDCIL dataset.

Table 1: Details of dataset used.

Dataset	Data split	Train	Dev	Test	Total
ULCA	Hours Utterances	8 3649	1 855	1 945	10 5449
LDCIL	Hours Utterances	8 8579	1 1200	1 1200	10 10979
ULCA+LDCIL	Hours Utterances	16 12228	2 2055	2 2145	20 16428

Table 2: Dialectal utterances count in ULCA+LDCIL

Dialects	#Train	#Dev	#Test
Imphal	6495	1370	1280
Kakching	2877	426	425
Sekmai	2750	401	400

The attention mechanism and Connectionist Temporal Classification (CTC) are the two main ASR approaches used by hybrid CTC/Attention-based encoder-decoder models. Two encoder architectures, namely the Transformer and the Conformer, are investigated. Transformers captures global acoustic contextual information using the attention mechanism. Instead of relying on recurrent connections like Recurrent Neural Networks or Long Short-Term Memory, Transformers use attention mechanisms to directly focus on the relevant parts of the input sequence, making them highly effective for sequential tasks like ASR. The Conformer in addition to utilizing convolutional layers to capture local contexts, which is particularly important for speech, also makes use of the attention mechanism to capture global information. Conformers combines the local modeling strengths of Convolutional Neural Networks (CNNs) with the global modeling capabilities of Transformers. Specifically optimized for tasks requiring both short-range and long-range dependency modeling. All models are trained using a transformer decoder [23]. For the front-end module, SSL models were used. SSL is a machine learning paradigm that learns representations from unannotated data by generating pseudo-labels. In speech recognition, SSL methods leverage large amounts of unlabeled speech data to learn representations that can be transferred to downstream ASR tasks. By applying SSL

Journal of Information Systems Engineering and Management
2025, 10(58s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

feature representations, ASR models can achieve better performance even with a small amount of labeled data, making it highly effective for low-resource languages. The SSL models used in this work are as follows:

• **Wav2Vec2 [24]:** A prominent SSL model for speech recognition, wav2vec 2.0 leverages contrastive learning to pretrain on large amounts of unlabeled speech data. The model learns to predict future speech segments from past ones, producing high-quality representations that can be fine-tuned on small labeled datasets for ASR tasks. This model has shown significant improvements in ASR performance for low-resource languages, even with limited labeled data.

• **HuBERT [25]:** The Hidden-Unit BERT (HuBERT) model is another SSL approach for speech recognition. It uses a clustering-based objective to learn discrete speech representations, which are then used for downstream ASR tasks. HuBERT has demonstrated success in handling dialects and phonetic variation in resource-scarce settings.

• **WavLM [26]:** Wavform based Language Model (WavLM), similar to Wav2Vec, is pretrained on large-scale unlabeled speech data but incorporates multilingual and diverse speech sources to improve robustness. It is designed to generalize better across various conditions, including noisy and reverberant environments.

• **XLSR-53 [27]:** XLSR-53 (eXtended Large Speech Representation 53) is a multilingual extension of the Wav2Vec 2.0 model, designed to handle multiple languages and work effectively across different speech tasks. It excels in multilingual speech recognition, enabling cross-lingual transfer and effective ASR for a wide range of languages, particularly where multilingual capabilities are needed.

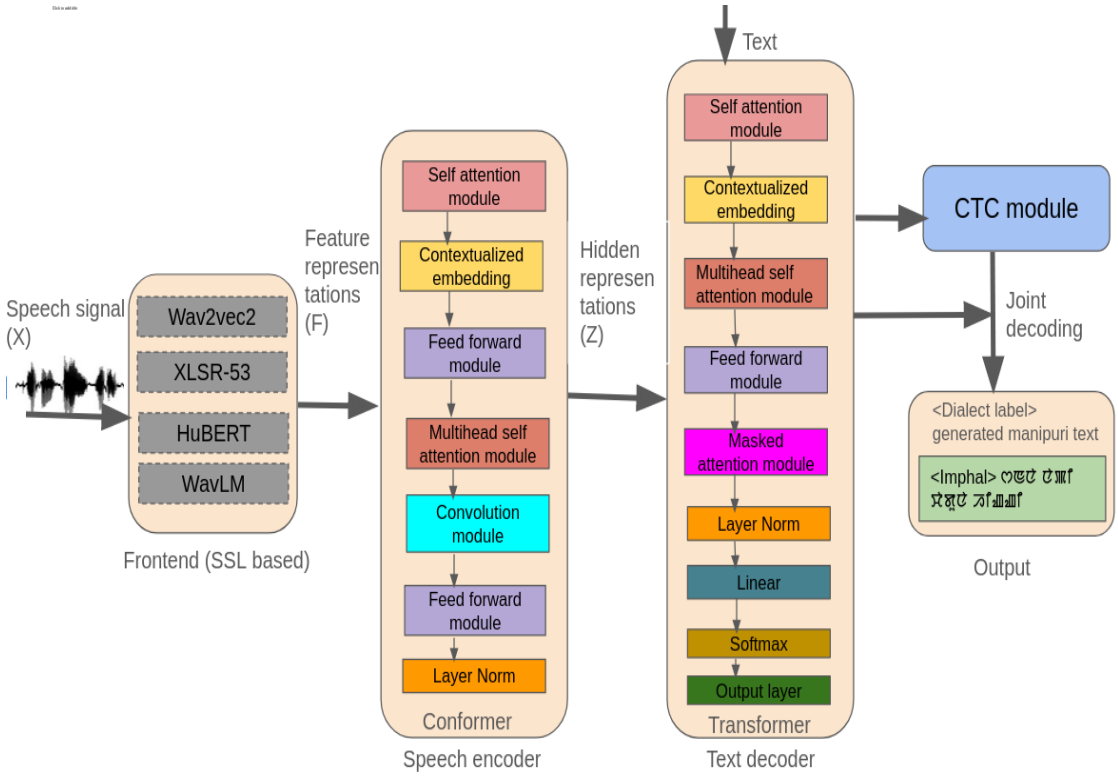

Figure 1. Model architecture of the proposed system performing Dialect Identification and Automatic Speech Recognition.

Figure 1 shows the overview of model architecture of the proposed system performing Dialect Identification and Automatic Speech Recognition. For each input manipuri speech signal (X), the frontend module (SSL based) is used extract feature representations (F). These feature representations serve as the input to the speech encoder. The speech encoder is responsible for taking the feature representations and applying self attention technique to convert them into contextualized embeddings that are dynamically generated based on the surrounding context of the tokens within a sentence. Multi head attention technique to focus on specific parts of the features. Encoders also employs

Journal of Information Systems Engineering and Management
2025, 10(58s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

feed forward neural networks using Rectified Linear Units (ReLU) for non-linear transformations which in turn helps the model to represent complex patterns in the input speech. One of the key component is the convolution module which effectively models local context in speech signals.

Algorithm 1: Algorithm for the proposed model

Require: Manipuri speech signals ($X = x_1 , x_2 , ...x_T$)

Ensure: Transcribed Manipuri text sequence ($Y = y_1 , y_2 , ...y_T$)

1: Extract the feature representations from the speech input through SSL based Frontend: $F = Frontend (X)$

2: Pass the features 'F' through the speech encoder (i.e., Conformer or Transformer) to get encoded/hidden representations: $Z = Encoder_{speech} (F)$

3: Pass the outputs of the encoder network to the Text Decoder (i.e., Transformer) to obtain the attention loss:

$L_{attention} = - \Sigma \log P (w_t |Z, w_{1:t-1})$ where $w_{1;t-1}$ is the ground truth of the previous word tokens.

4: Integrate the outputs of the Text decoder network with the CTC model to obtain the CTC loss:

$L_{CTC} = -\log (P (W|Z))$ where 'W' is the target sequence.

5: Combine the CTC loss and the attention loss to get the total train loss:

$L_{Total} = \alpha * L_{CTC} + (1 - \alpha) * L_{attention}$ where α is the CTC weight.

6: Optimize the model by minimizing the total loss over the training data:

$\theta_{model} = argmin (L_{Total})$.

7: For test data speech, extract the frontend SSL features, encode them with speech encoder and use joint CTC-attention decoding to the manipuri corresponding transcription and return the transcribed output: W_{test}.

The text decoder, on the other hand, generates the dialect label and the manipuri text transcriptions. The output from the speech encoder, i.e., hidden representations (Z) is passed through the text decoder and similar techniques like self attention and feed forward modules are applied. During the training process, the decoder applies masked attention technique so that it does not use the future information while generating the output tokens. Output tokens are generated based on the current input and previously generated tokens. The decoder uses an auto-regressive technique to predict the text output. It creates a probability distribution across the target vocabulary by using linear layers and a soft-max activation function. For a given input, this allows the model to produce the most likely string of tokens. The output that is produced sometimes could not match the ground truth transcription exactly. By taking into account potential alignments and variances in input speech, CTC module assists in matching the generated sequence with the ground truth transcription. The weighted sum of the CTC loss and the decoder loss is used to optimize the model during training. By employing a beam search to jointly decode the encoder and decoder's output, the final hypothesis is produced during inference. In order to improve ASR performance, this hybrid technique successfully blends the contextual sensitivity of the attention mechanism with the strong alignment capabilities of CTC[28]. Algorithm 1 for the proposed model discusses the same. For the task of Dialect identification, the above model structure is used in which target dialect is generated explicitly in the text output sequence, by adding a dialect label at the beginning to the text to be predicted. The vocabulary of the ASR model is extended to include dialect labels in the following way:

$$'V = V \cup \{ [Imphal] , [Kakching] , [Sekmai] \} \quad (1)$$

By inserting the dialect label at the beginning, it first predicts a dialect label and then performs speech recognition conditioned on the predicted label, similar to having a dialect identification module as pre-processing without actually requiring an additional dialect identification module. This approach efficiently jointly models the ASR task and the dialect identification task.

Journal of Information Systems Engineering and Management
2025, 10(58s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

All our experiments have been implemented using Espnet toolkit [29]. Figure 2 shows the steps involved in the implementation of the method.

• **Data preparation:** In this stage, the entire dataset is split into train, dev and test, and prepared them in the Kaldi [30] format. After this part is finished, the newly created directories i.e.,train, dev and test will be available. Each of these directories will have four different files: spk2utt (Speaker information), text (Transcription file), utt2spk (Speaker information) and wav.scp (Audio file). These are the speech and corresponding text and speaker information in the Kaldi format. Here, the number of utterances for train set was 12126, dev set with 2197 and test set with 2105.

• **Speed perturbation:** Speed perturbation is performed and then save the augmented data in the disk before training on the train set. Another approach is to perform data augmentation during training, such as SpecAug. Both speed perturbation and SpecAug have been used in this work. For speed perturbation, factors 0.9, 1.0 and 1.1 was used. After this step is completed, the number of utterances in the train set was increased by three folds i.e., 36378. No changes were done for dev and test set.

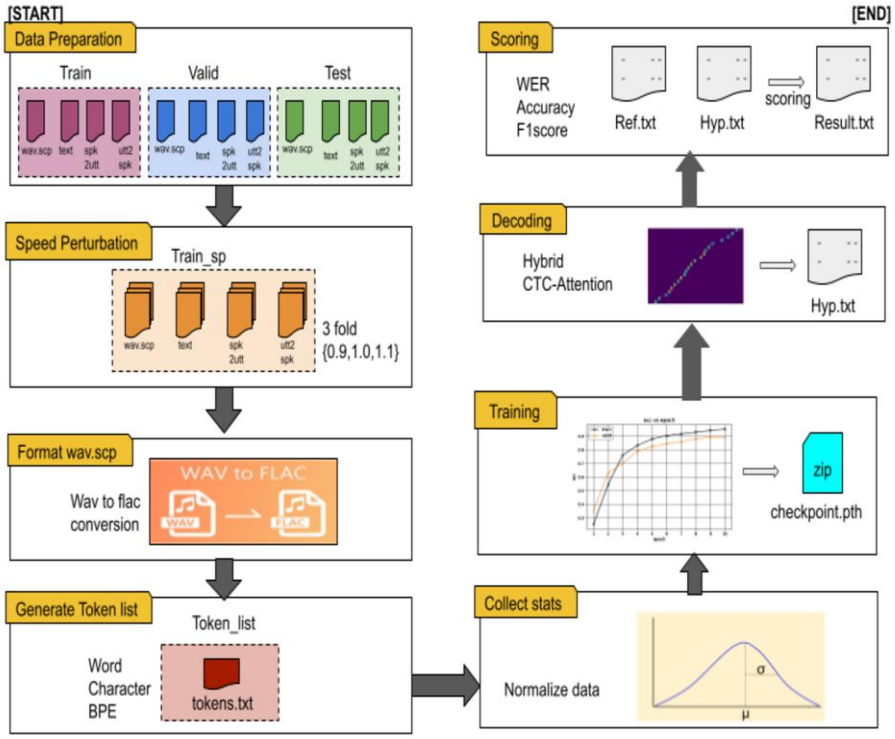

Figure 2. Implementation Steps of Multi-task learning systems performing Dialect Identification and Automatic Speech Recognition.

• **Format wav.scp:** Formatting of the data with specified format (flac in this case) is done for the efficient use of the data. This is done on train and dev data. Also, too long and too short audio data are harmful for efficient training. Those utterances are removed for training. But for inference and scoring, still the full data is used, which is important for fair comparison.

• **Generate token list:** Next the token list is generated from the train set. This is important for text processing. Here, a dictionary is prepared simply using the Manipuri characters. The list can be generated based on word , character or Byte Pair Encoding (BPE) type. BPE provides a middle ground between word-based and character-based tokenization. The sentencepiece toolkit developed by Google is used. For word type, the total number of unique tokens in the token list were 12185. For BPE type, 500 BPE tokens are used.

Journal of Information Systems Engineering and Management
2025, 10(58s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

• **Collect stats:** Then, estimated the mean and variance of the data to normalize the data. The information of input and output lengths is also collected for the efficient mini batch creation shown as a scatter plot of the same in Figure 3. It shows the distribution of each utterance in three different classes of dialect.

• **Training:** The seven SSL models namely, Wav2Vev2(Base), Wav2Vec2(Large), HuBERT(Base), HuBERT(Large), WavLM(Base), WavLM(Large), XLSR-53 provided by s3prl framework supported by Espnet is used for our experiments. For building these models, a single NVIDIA GPU Tesla V100 16GB is used. The model hyperparameters were kept the same for all our models configurations as provided in the Table 3. The Adam optimizer is used to adjust the weights of the model during training. Learning rate controls the size of the steps that the optimization algorithm takes when adjusting the weights of the model towards the minimum of the loss function. Three different learning rates were used i.e., 5e-3, 5e-4, 5e-5 for different models. 10k and 20k warmup steps were

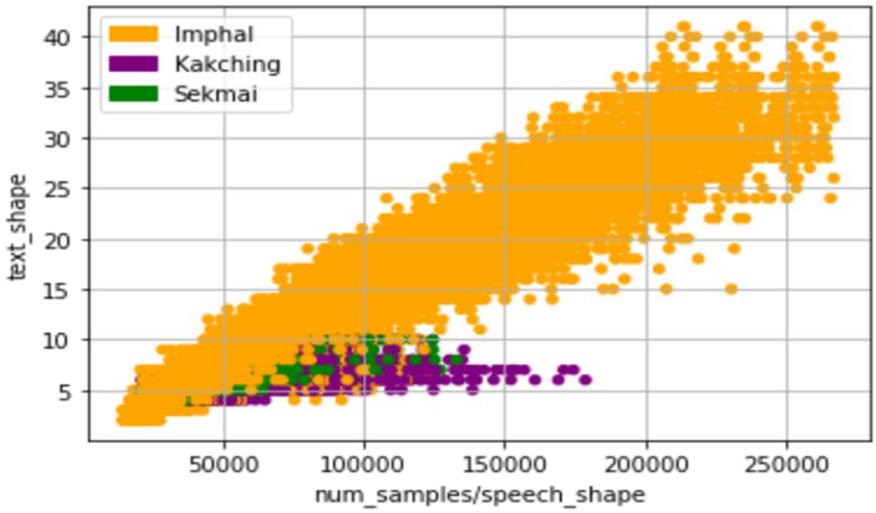

Figure 3. Scatter plot of number of words vs number of utterance samples for each utterance. Colors correspond to Imphal dialect (orange), Kakching dialect (purple) and Sekmai dialect (green)

Table 3: Hyperparameters used in the implementation

Hyperparameters	Values for word type	Values for BPE type
Optimizer	Adam	Adam
Learning rate	5e-3, 5e-4, 5e-5	5e-4
warmup steps	10k,20k	10k
epoch	10	10
batch size	32,64	32,64

Journal of Information Systems Engineering and Management
2025, 10(58s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

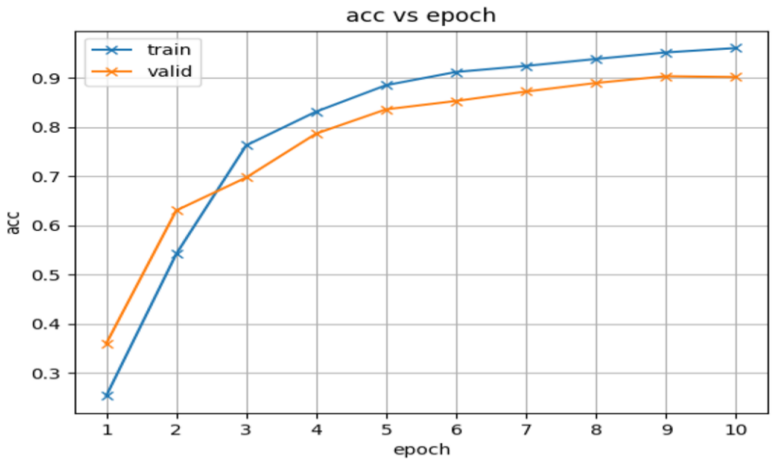

Figure 4. Line plot of the training and dev accuracy values over several training epochs when the dialect information is used with BPE based ASR.

chosen. The number of epochs during the training was set to 10, due to the complexity of the task and the small size of the labeled dataset. The batch size controls how many training instances are fed into the model at the same time during training. Several values were tested in order to achieve better results. Batch sizes of 32 and 64 were used in the experiment. More details about the configuration file related to this work and all the training checkpoints will be made available on our GitHub page and can be used directly with the model implementation from Espnet. The training progress is monitored by checking the dev score. Figure 4. shows the accuracy vs number of epochs plot on train and dev data generated during the training. The plots shown are of BPE based Conformer models with dialect information. It gives us better insight into how the training performance changes over the number of epochs. When analyzing the learning curve of the model, it is observable that the accuracy value increases, converging at each training epoch. However, values greater than 10 epochs did not provide significant higher values in the accuracy. This suggests that the model has achieved considerable stability and does not substantially benefit from additional training. The training is stopped in order to optimize training resources and saving computational time.

• **Decoding:** After the model is trained, it is used to decode the utterances of the test set to generate the corresponding text. The CTC weight 0.5 was used. Beam size was set to 10. The output here is the hypothesis file with the generated sentences.

• **Scoring:** The performance of the model are measured in terms of word error rate (WER), character error rate (CER), token error rate (TER) for ASR. Accuracy, Precision, Recall and F1 score for dialect identification. Each sentences in the reference (ground truth) file is compared with hypothesis file and scores are calculated and stored in the result file.

RESULTS

This section presents the results arising from the application of the model developed to address the task of dialect identification and ASR. Throughout this work, a variety of approaches were explored empirically to enhance the model's ability to correctly generate the corresponding texts from the given utterance on test data, with the aim of contributing to the advancement of speech technologies. Evaluating the performance of an Automatic Speech Recognition (ASR) system is crucial to understanding its accuracy, efficiency, and practical applicability. Standard performance measures have been used for checking the performance of Dialect identification and ASR systems [31].

Metrics used for evaluating the dialect identification process are as follows:

• **Accuracy:** Measures the overall correctness of a model's predictions. It is calculated by dividing the number of correct predictions by the total number of predictions.

$$Accuracy = \frac{TruePositive_{overall} + TrueNegative_{overall}}{TotalPredictions} \quad (2)$$

• **Precision:** This measures the proportion of correctly predicted positive instances (e.g., correctly detected words or phrases) out of all predicted positive instances.

Journal of Information Systems Engineering and Management
2025, 10(58s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

$$Precision = \frac{TruePositive_{overall}}{TruePositive_{overall}+FalsePositive_{overall}} \quad (3)$$

• **Recall:** Recall measures the proportion of correctly predicted positive instances out of all actual positive instances.

$$Recall = \frac{TruePositive_{overall}}{TruePositive_{overall}+FalseNegative_{overall}} \quad (4)$$

• **F1 score:** F1 Score is the harmonic mean of Precision and Recall and provides a single metric that balances the two.

$$F1score = \frac{2*Precision*Recall}{Precision+Recall} \quad (5)$$

A higher F1 Score indicates a better balance between precision and recall. F1score is preferable to accuracy for class-imbalanced datasets.

Two metrics are used to evaluate ASR systems, each providing insight into different aspects of performance. Below are some of the key evaluation metrics commonly used in ASR:

• **Word Error Rate (WER):** WER is the most commonly used metric to evaluate the performance of ASR systems, especially in general-purpose applications. It measures how many words were incorrectly recognized compared to the ground truth.

$$WER = \frac{S+D+I}{N} \quad (6)$$

where, S is the number of substitutions (wrong words), D is the number of deletions (missed words), I is the number of insertions (extra words), N is the total number of words in the reference transcript (ground truth). A lower WER indicates better ASR performance.

• **Real-Time Factor (RTF):** The RTF measures the speed of the ASR system in relation to the duration of the input audio. It is particularly important for real-time speech recognition systems. It tells us how many times slower the system is compared to the real-time audio playback.

$$RTF = \frac{TimeTakenForProcessing}{AudioLength} \quad (7)$$

An RTF of 1.0 means the system processes audio in real-time. An RTF greater than 1.0 means the system is slower than real-time. An RTF less than 1.0 means the system is faster than real-time.

Table 4 and Table 5 gives the performance measures in terms of accuracy, precision, recall and F1score for dialect identification when Transformer and Conformer based ASR are used respectively. The highest accuracy = 87.26% by Conformer with Wav2Vec2(L), precision = 89.70% by Transformer with Wav2Vec2(B), Recall = 87.26% by Conformer with Wav2Vec2(L) and F1score = 86.27% by Conformer with Wav2Vec2(L).

Table 4: Dialect recognition results when BPE is used for Transformer based ASR is used (in percentage)

SSL Type	Dev				Test			
	Acc	Prec	Recall	F1score	Acc	Prec	Recall	F1score
Wav2Vec2(B)	92.89	93.38	92.89	**93.11**	86.08	89.70	86.08	**85.01**
Wav2Vec2(L)	91.35	92.31	91.35	91.66	85.36	87.52	85.36	84.83
HuBERT(B)	91.67	92.01	91.67	91.79	83.04	85.11	83.04	80.92
HuBERT(L)	92.35	93.02	92.35	92.60	84.51	85.60	84.51	82.98
WavLM(B)	90.48	91.23	90.48	90.75	82.80	85.70	82.80	81.16

Journal of Information Systems Engineering and Management
2025, 10(58s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

WavLM(L)	90.80	91.74	90.80	91.10	83.32	86.14	83.32	82.51
XLSR-53	89.71	91.50	89.71	90.06	80.66	83.40	80.66	79.28

Table 5: Dialect recognition results when BPE is used for Conformer based ASR is used (in percentage)

SSL Type	**Dev**				**Test**			
	Acc	**Prec**	**Recall**	**F1score**	**Acc**	**Prec**	**Recall**	**F1score**
Wav2Vec2(B)	93.21	93.56	93.21	93.31	83.08	85.85	83.08	81.54
Wav2Vec2(L)	92.85	92.90	92.85	92.79	87.26	87.78	87.26	**86.27**
HuBERT(B)	92.39	92.43	92.39	92.34	85.03	86.80	85.03	82.82
HuBERT(L)	93.03	93.44	93.03	93.08	81.99	84.05	81.99	78.38
WavLM(B)	90.89	91.73	90.89	91.12	82.04	84.37	82.04	80.59
WavLM(L)	93.30	93.69	93.30	**93.33**	83.18	86.82	83.18	80.02
XLSR-53	91.44	92.01	91.44	91.63	84.46	86.94	84.46	83.23

The two confusion matrices for the model with the highest F1score on test data i.e., Conformer with Wav2Vec2(L) are represented as a heat map for better graphical representation shown in Figure 5 for both dev and test data. This heat map can conveniently display the classification and misclassification of our three classes of dialects. From this confusion matrix in Figure 5(a), it is observed that on dev data, Imphal dialect is misclassified only 3% of times with other dialects, Kakching dialect is misclassified almost 18% of times with other dialects and Sekmai dialect is misclassified almost 10% of times with other dialects. Figure 5(b) shows on test data, the misclassi-fication of Imphal dialect is around 4% of times with other dialects. Kakching dialect is misclassified 6% of times with others and Sekmai dialect is misclassified 50% of times with other dialects.

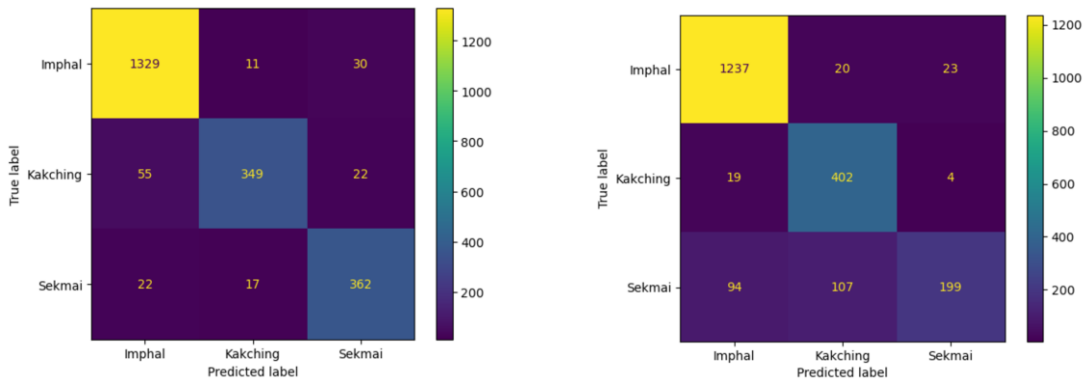

Figure 5. Confusion matrix for the model with the highest F1score (86.27%) achieved on test data i.e., Conformer with Wav2Vec2(L).

Results from experiments are presented in Tables – 6, 7 and 8 for ASR, with lowest WERs highlighted in bold. Both Transformer based and Conformer based models have used 7 different types of SSL features. Table 6 gives the ASR

Journal of Information Systems Engineering and Management
2025, 10(58s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

result without using any dialect information (word based), in terms of WER and CER on dev and test data. From this table, it is observable that the performance of all models is poor except for the Transformer model with HuBERT(L) with 47.4% WER and 38.7% CER. These results provide a baseline where no dialect information is used and no dialect identification was performed.

Table 6: Word based ASR Results without using any dialect information in terms of WER and CER (in percentage)

Model	SSL Type	Dev		Test	
		WER (↓)	CER(↓)	WER (↓)	CER(↓)
Transformer	Wav2Vec2(B)	62.3	49.7	57.3	46.6
	Wav2Vec2(L)	62.2	48.4	57.3	45.6
	HuBERT(B)	76.6	59.5	68.4	55.2
	HuBERT(L)	49.8	40.3	47.4	38.7
	WavLM(B)	69.5	57.8	66.0	56.3
	WavLM(L)	53.9	42.5	50.5	40.4
	XLSR-53	100.7	82.5	101.5	84.5
Conformer	Wav2Vec2(B)	65.1	52.9	61.6	50.9
	Wav2Vec2(L)	66.1	53.7	62.2	51.9
	HuBERT(B)	68.5	55.5	63.6	53.8
	HuBERT(L)	64.9	53.6	62.2	52.1
	WavLM(B)	69.2	56.5	65.0	55.4
	WavLM(L)	63.6	50.1	58.1	46.9
	XLSR-53	70.3	56.8	67.0	55.5

Next, after introducing the dialect information, a little improvement can be seen in the results. Table 7 shows the ASR result using dialect information (word based), in terms of WER and CER on dev and test data. The best result in this case was given again by the Transformer model with HuBERT(L) with 42.0% WER and 32.6% CER on test data. The objective was to investigate whether the addition of the dialect information during training has any impact on the model performance. All models achieved lower WERs on dev and test data comparing to Table 6 experiments.

Table 7: Word based ASR Results using dialect information in terms of WER and CER (in percentage)

Model	SSL Type	Dev		Test	
		WER (↓)	CER(↓)	WER (↓)	CER(↓)
Transformer	Wav2Vec2(B)	54.9	42.1	51.4	40.1
	Wav2Vec2(L)	54.6	41.5	50.9	39.1
	HuBERT(B)	54.4	41.2	51.0	39.6
	HuBERT(L)	44.0	33.9	42.0	32.6
	WavLM(B)	57.2	43.7	52.8	41.0
	WavLM(L)	46.3	35.1	44.0	33.8
	XLSR-53	58.6	45.7	54.2	43.0
Conformer	Wav2Vec2(B)	59.0	46.5	56.5	45.1
	Wav2Vec2(L)	58.8	44.7	54.5	42.6
	HuBERT(B)	57.9	43.3	53.0	40.9
	HuBERT(L)	54.6	42.8	52.1	41.2
	WavLM(B)	59.1	44.1	54.3	42.1
	WavLM(L)	52.4	40.1	48.8	38.1
	XLSR-53	61.5	47.4	57.1	45.4

Journal of Information Systems Engineering and Management
2025, 10(58s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

With the aim of further improving the ASR results, similar experiments with BPE tokenization was performed as shown in Table 8. In this case, all models have outperformed the models in the previous experiments and gave the best WER on dev and test data. There is a significant improvement in the results. Best WER and CER are 19.8% and 6.6% respectively. This result was achieved from using Conformer based model with WavLM(L). All Conformer based models outperformed the transformer based models. Performance of models from previous experiments in comparison with models with BPE based highlights the limitation of using a fixed vocabulary instead of an open vocabulary. This may imply that open vocabulary might have helped the model to recognize words that were never seen during the training process.

Table 8: BPE based ASR Results using dialect information in terms of WER and CER (in percentage)

Model	**SSL Type**	**Dev**		**Test**	
		WER (↓)	**CER(↓)**	**WER (↓)**	**CER(↓)**
Transformer	Wav2Vec2(B)	38.4	17.0	35.8	16.6
	Wav2Vec2(L)	36.5	15.1	34.0	14.5
	HuBERT(B)	37.4	16.0	34.9	15.8
	HuBERT(L)	27.3	9.1	25.7	9.4
	WavLM(B)	36.8	15.3	34.5	15.1
	WavLM(L)	25.1	7.9	24.0	8.4
	XLSR-53	32.0	12.5	30.0	12.3
Conformer	Wav2Vec2(B)	25.3	8.6	23.9	8.9
	Wav2Vec2(L)	22.8	7.3	21.7	7.5
	HuBERT(B)	17.5	7.8	22.9	8.2
	HuBERT(L)	22.1	6.6	21.6	7.3
	WavLM(B)	24.0	7.9	23.0	8.4
	WavLM(L)	19.9	5.8	**19.8**	**6.6**
	XLSR-53	29.9	12.3	28.5	12.3

Next, the RTF is also calculated for all models. Figure 6 shows the RTF bar plots with different SSL types. For word-based ASR, the best RTF value and the worst RTF value are 0.082 by transformer with HuBERT(L) and 0.120 by conformer with WavLM(L) respectively. For BPE based ASR, the best RTF value and the worst RTF value are 0.161 by Conformer with HuBERT(B) and 0.230 by Conformer with HuBERT(L) respectively.

Journal of Information Systems Engineering and Management
2025, 10(58s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

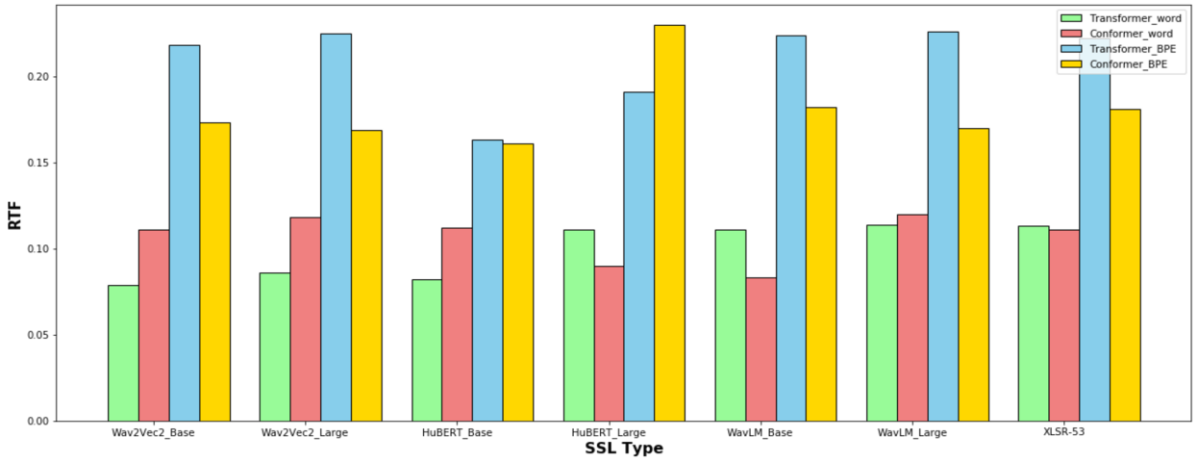

Figure 6. RTF plot for the models.

DISCUSSION

1. Table 6 provides a baseline result for ASR when no dialect information is used during the training. The WER and CER results are not impressive but it was found that HuBERT(L) shows a descent result out of all the other models. There is a variation in WER and CER between (L) and (B) type models across all the test set, so the (L) models are giving better results than (B) models in the low resource scenario.

2. Next, after including the dialect information during training, in Table 7, it is noted that there is a consistent improvement in WER and CER results for all models. This indicates that the joint training of dialect identification and ASR helps improve the ASR result.

3. Then, in Table 8, where the subword-based tokenization with dialect information is used, it was found that there is a further improvement and gives the best results with WER and CER of 19.8% and 6.6% on the test sets respectively. This might be due to better handling of rare and/or unseen words in the test set.

4. In Figure 6, it was noticed that the RTF values are lower for word based ASR than BPE based i.e., the decoding time for test data is faster for word based than for BPE based. The reason being during BPE-based decoding process, the smaller subword units or tokens are concatenated together to form original word while word-based decoding does not require any subword concatenation as the whole word is treated as an individual unit or token.

5. Figure 5 confusion matrix for both dev and test data. The classification result for Sekmai dialect class is poor and highlights a possiblity of noisy utterances for this particular class data. Further investigation is needed, however it might require much cleaner and more data.

6. To the best of our knowledge, no previous work related to Dialect identification and ASR exists on this particular dataset. With the use of SSL models and dialect information, the SOTA results are obtained and can be set as a benchmark for this dataset.

7. While this work provides valuable insights, there are limitations to our study. The model was trained on the news domain and common sentences speech data. Its performance in out-of-domain speech remains invalidated. Also, the suitability of the models for noisy speech data was not evaluated.

This work in the context of previous Manipuri ASR approaches:

Previous authors have adopted various approaches to build Manipuri ASR systems but on a different datasets or on a common subpart, thus it would not be fair to make any direct comparisons.

Journal of Information Systems Engineering and Management
2025, 10(58s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Table 9: Comparison in context of previous Manipuri ASR

Paper	Year	Dataset	Features	Technique	WER(in %)
[17]	2018	Read speech, lectures and spontaneous data	PLP and MFCC	GMM-HMM	----
[18]	2018	Telephonic speech data	MFCC	DNN-HMM	13.57
[19]	2024	MECOS	MFCC and i-vectors	TDNN	33.17
[20]	2025	ULCA	Log mel features and wav2vec2	Conformer	29.7
This work	2025	ULCA+LDCIL	SSL features	Conformer in a multitask framework	19.8

Our best results using the SSL features show potential for significant improvement over the previously reported results on the common subpart dataset as shown in Table 9. Our trained models achieved the best WER of 19.8 on the ULCA+LDCIL dataset as compared to the previously reported results from [18], [19], and [20]. Our work can be set as a benchmark for this particular dataset and should further provide researchers with a useful basis for future comparisons.

CONCLUSION

While the field of speech recognition for Manipuri is in its infancy, recent advancements in machine learning and natural language processing offer significant promise. Addressing the challenges posed by dialectal diversity, phonetic complexity, and data scarcity will require interdisciplinary efforts combining linguistics, computational linguistics, and machine learning. The future of ASR for Manipuri holds exciting potential, particularly with the increasing availability of data and improvements in computational power. By overcoming these challenges, it is possible to create inclusive and effective speech recognition systems that serve the diverse linguistic community of Manipuri speakers. In this work, both Transformer and Conformer based encoders with different SSL features were investigated to develop both dialect identification and ASR systems. Experiments were designed to evaluate the relative performance of the models with and without using dialect information. Using dialect information, the ASR results did improved. And further improvement was achieved by using BPE based tokenization instead of word based ASR. The best WER and CER on test data are 19.8% and 6.6% respectively, and can be set as SOTA result on this particular dataset. Future work may involve transcribing additional Manipuri unlabeled speech data using these models. Investigation of unsupervised approach using GAN for low resource languages can also be explored.

REFERENCES

[1] Besacier, L., Barnard, E., Karpov, A., Schultz, T.: *Automatic speech recognition for under-resourced languages: A survey*. Speech communication 56 , 85–100 (2014)
[2] Imaizumi, R., Masumura, R., Shiota, S., Kiya, H.: *Dialect-aware modeling for end-to-end japanese dialect speech recognition*. In: 2020 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPAASC), pp. 297–301 (2020). IEEE
[3] *Baevski, A., Mohamed, A.: Effectiveness of self-supervised pre-training for asr*. In: ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 7694–7698 (2020). IEEE
[4] Sailor, H.B., Patil, A.T., Patil, H.A.: *Advances in low resource asr: A deep learning perspective*. In: SLTU, pp. 15–19 (2018)

[5] Zevallos, R.: *Text-to-speech data augmentation for low resource speech recognition*. arXiv preprint arXiv:2204.00291 (2022)
[6] Thai, B., Jimerson, R., Arcoraci, D., Prud'hommeaux, E., Ptucha, R.: *Synthetic data augmentation for improving low-resource asr*. In: 2019 IEEE Western New York Image and Signal Processing Workshop (WNYISPW), pp. 1–9 (2019). IEEE
[7] Tüske, Z., Golik, P., Nolden, D., Schlüter, R., Ney, H.: *Data augmentation, feature combination, and multilingual neural networks to improve asr and kws performance for low-resource languages*. In: Interspeech, pp. 1420–1424 (2014)
[8] Anoop, C., Ramakrishnan, A.: *Ctc-based end-to-end asr for the low resource sanskrit language with spectrogram augmentation*. In: 2021 National Conference on Communications (NCC), pp. 1–6 (2021). IEEE
[9] Asadolahzade, M.: *Transfer learning for asr to deal with low-resource data problem*. Tehran, Iran, May (2019)
[10] Bataev, V., Korenevsky, M., Medennikov, I., Zatvornitskiy, A.: *Exploring end-to-end techniques for low-resource speech recognition*. In: Speech and Computer: 20th International Conference, SPECOM 2018, Leipzig, Germany, September 18–22, 2018, Proceedings 20, pp. 32–41 (2018). Springer
[11] Deng, K., Cao, S., Ma, L.: *Improving accent identification and accented speech recognition under a framework of self-supervised learning*. arXiv preprint arXiv:2109.07349 (2021)
[12] Safieh, A.A., Alhaol, I.A., Ghnemat, R.: *End-to-end jordanian dialect speech-to-text self-supervised learning framework*. Frontiers in Robotics and AI 9 , 1090012 (2022)
[13] Miwa, S., Kai, A.: *Dialect speech recognition modeling using corpus of japanese dialects and self-supervised learning-based model xlsr*. Proc. INTERSPEECH 2023, 4928–4932 (2023)
[14] Caubrière, A., Gauthier, E.: *Africa-centric self-supervised pre-training for multilingual speech representation in a sub-saharan context*. arXiv preprint arXiv:2404.02000 (2024)
[15] Mdhaffar, S., Elleuch, H., Bougares, F., Estève, Y.: *Performance analysis of speech encoders for low-resource slu and asr in tunisian dialect*. arXiv preprint arXiv:2407.04533 (2024)
[16] Lodagala, V.S., Biswas, A., Das, S., Umesh, S., et al. : *All ears: Building self-supervised learning based asr models for indian languages at scale*. In: Proc. Interspeech 2024, pp. 3944–3948 (2024)
[17] Dutta, S.K., Singh, L.J.: *A comparison of three spectral features for phone recognition in sub-optimal environments*. International Journal of Applied Pattern Recognition 5 (2), 137–148 (2018)
[18] Patel, T., Krishna, D., Fathima, N., Shah, N., Mahima, C., Kumar, D., Iyengar, A.: *Development of large vocabulary speech recognition system with keyword search for manipuri*. In: Interspeech, pp. 1031–1035 (2018)
[19] Singh, N.K., Chanu, Y.J., Pangsatabam, H.: *Mecos: A bilingual manipuri–english spontaneous code-switching speech corpus for automatic speech recognition*. Computer Speech & Language 87 , 101627 (2024)
[20] Devi, T.C., Thaoroijam, K., Nongmeikapam, K.: *Improving conformer based end- to-end manipuri automatic speech recognition using wav2vec2 model*. Journal of Information Systems Engineering and Management 10 (56) (2025)
[21] Chadha, H.S., Gupta, A., Shah, P., Chhimwal, N., Dhuriya, A., Gaur, R., Raghavan, V.: *Vakyansh: Asr toolkit for low resource indic languages*. arXiv preprint arXiv:2203.16512 (2022)
[22] Choudhary, N., Rao, D.: *The ldc-il speech corpora*. In: 2020 23rd Conference of the Oriental COCOSDA International Committee for the Co-ordination and Standardisation of Speech Databases and Assessment Techniques (O-COCOSDA), pp. 28–32 (2020). IEEE
[23] Gulati, A., Qin, J., Chiu, C.C., Parmar, N., Zhang, Y., Yu, J., Han, W., Wang, S., Zhang, Z., Wu, Y., et al.: *Conformer: Convolution-augmented transformer for speech recognition*. arXiv preprint arXiv:2005.08100 (2020)
[24] Baevski, A., Zhou, Y., Mohamed, A., Auli, M.: *wav2vec 2.0: A framework for self-supervised learning of speech representations*. Advances in neural information processing systems 33 , 12449–12460 (2020)
[25] Hsu, W.-N., Bolte, B., Tsai, Y.-H.H., Lakhotia, K., Salakhutdinov, R., Mohamed, A.: *Hubert: Self-supervised speech representation learning by masked prediction of hidden units*. IEEE/ACM transactions on audio, speech, and language processing 29 , 3451–3460 (2021)
[26] Chen, S., Wang, C., Chen, Z., Wu, Y., Liu, S., Chen, Z., Li, J., Kanda, N., Yoshioka, T., Xiao, X., et al. : *Wavlm: Large-scale self-supervised pre-training for full stack speech processing*. IEEE Journal of Selected Topics in

Signal Processing 16 (6), 1505–1518 (2022)
[27] Conneau, A., Baevski, A., Collobert, R., Mohamed, A., Auli, M.: *Unsupervised cross-lingual representation learning for speech recognition*. arXiv preprint arXiv:2006.13979 (2020)
[28] Watanabe, S., Hori, T., Kim, S., Hershey, J.R., Hayashi, T.: *Hybrid ctc/attention architecture for end-to-end speech recognition*. IEEE Journal of Selected Topics in Signal Processing 11 (8), 1240–1253 (2017)
[29] Watanabe, S., Boyer, F., Chang, X., Guo, P., Hayashi, T., Higuchi, Y., Hori, T., Huang, W.C., Inaguma, H., Kamo, N., et al. : *The 2020 espnet update: new features, broadened applications, performance improvements, and future plans*. In: 2021 IEEE Data Science and Learning Workshop (DSLW), pp. 1–6 (2021). IEEE
[30] Ravanelli, M., Parcollet, T., Bengio, Y.: *The pytorch-kaldi speech recognition toolkit*. In: ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 6465–6469 (2019). IEEE
[31] Martin, J.H.: *Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition* (2009)