Active Learning Approach for Fine-Tuning Pre-Trained ASR Model for a low-resourced Language

Rupak Raj Ghimire, Bal Krishna Bal and Prakash Poudyal

Information and Language Processing Research Lab (ILPRL)
Department of Computer Science and Engineering
Kathmandu University, Nepal

rughimire@gmail.com, {bal, prakash}@ku.edu.np

Abstract

Fine tuning of the pre-trained language model is a technique which can be used to enhance the technologies of low-resourced languages. The unsupervised approach can fine-tune any pre-trained model with minimum or even no language-specific resources. It is highly advantageous, particularly for languages that possess limited computational resources.

We present a novel approach for fine-tuning a pre-trained Automatic Speech Recognition (ASR) model that is suitable for low resource languages. Our methods involves iterative fine-tuning of pre-trained ASR model. *mms-1b* is selected as the pre-trained seed model for fine-tuning. We take the Nepali language as a case study for this research work. Our approach achieved a CER of 6.77%, outperforming all previously recorded CER values for the Nepali ASR Systems.

Keywords - Low-Resourced ASR, Active Learning, Pre-Training, Nepali ASR

1 Introduction

Automatic Speech Recognition (ASR) systems use machine learning algorithms and neural networks to analyze and transcribe audio recordings or live speech into text form. ASR has a wide range of applications, including - voice assistants, transcription services, accessibility tools, language translation services, voice search etc. ASR technology has advanced significantly in recent years following the development of deep learning techniques, and it continues to play a crucial role in the development of natural language processing and human-computer interaction systems.

The supervised ASR systems learn from aligned speech and transcribed text pairings, requiring lots of well-prepared data. Speech and transcribed text must be perfectly aligned and the alignment basically is done manually in the Supervised approach. The manual alignment demands extensive time and effort as spoken words must be precisely matched with their written counterparts. Unsupervised ASR implementation was first proposed by Liu et al. (2018). Since then unsupervised approaches have been in popular use. Recently, the research conducted by Baevski et al. (2022) demonstrated that the performance of unsupervised ASR is equivalent to supervised ASR models performance.

However, developing an ASR model for low-resourced languages is a challenging task. The pre-trained model could be fine-tuned with some limited available labeled speech corpus. This fine-tuned model is then used to label the unlabeled corpus. The labeling achieved can be augmented to the manually labeled speech corpus. The whole process can be described as a semi-supervised learning approach which is a combination of supervised and unsupervised approaches. Also, there is yet another approach called transfer learning which is also becoming predominantly popular in such contexts. Transfer learning as the name suggests is about applying a model trained and developed for one purpose or task to a somewhat related but quite different task and purpose. Similarly, active learning is focused on selecting the most informative examples for labeling to improve the model's performance since manual labeling is resource-intensive.

While they both aim to enhance machine learning models, they address different aspects of the learning process: transfer learning focuses on the use of other supportive models, while active learning concentrates on data selection and annotation.

The concept of the active learning are being popular in the ASR domain after popularity of the larger ASR pre-trained model such as HuBERT(Hsu et al., 2021), wav2vec(Schneider et al., 2019), wav2vec2.0(Baevski et al., 2020a, Baevski et al., 2020b, w2v-BERT(Chung et al., 2021), Mockingjay(Liu et al., 2020). These models are trained using larger and multilingual speech corpus. Many research (Arunkumar et al., 2022; Khare et al., 2021; Luo et al., 2021; Singh et al., 2023; Zheng et al., 2023) show that the accuracy of the ASR in low-resourced languages can be improved by fine-tuning the pre-trained models. The fine-tuning approach also saves the model training time significantly.

In this paper, we propose an active learning-based self-training approach to improve the quality of the ASR model. We used language score as a active learning dataset selection criteria. *wav2vec2.0* based Massively Multilingual Speech (MMS) - 1B (*mms-1b*) pre-trained ASR model is used as a seed model for fine tuning.

The rest of the paper is organized as follows the related works and the methodology used are explained in section 2 and 3 respectively. Section 4 explains the dataset used for training the model. Section 5 presents the conducted experiments and their results. Finally, the paper concludes with section 6 where summary of findings, future plans, and potential extensions to the work are explained.

2 Related Works

Self-supervised approaches have been popular for the past few years, especially in the context of low-resourced languages. The goal of this approach is to optimistically utilize the large unlabeled speech data to improve the overall accuracy of any given task in low-resourced languages. There are various large models available for ASR task. Hsu et al. (2021) has proposed a Hidden-Unit BERT (HuBERT), a self-supervised speech representation learning, which utilizes an offline clustering step to provide aligned target labels for a BERT-like prediction loss. Javed et al. (2022) combined various speech dataset of 40 indian languages (17,000 hours of raw speech data) including Nepali and trained wav2vec pre-trained model called *IndicWav2Vec*.

Baevski et al. (2020a) has proposed a framework for the semi-supervised learning of speech representation. The authors trained unsupervised model on 53K hours of unlabeled data to produce pre-trained seed model. That model is further fined tuned using just 10 minutes of labeled data. They achieved 4.8/8.2% WER on clear/other test sets. This demonstrates the feasibility of speech recognition with limited amounts of labeled data.

The data augmentation-based approach is another way to take benefit of the larger resources. Khare et al. (2021) utilized the transliteration based augmentation to address the labeled data scarcity. High-resource graphemes of English are transliterated to low-resourced languages graphemes(experimented on : Hindi, Telugu, Gujarati,Bengali, Korean, Amharic). Those transliterated dataset are used for extended pre-training of the seed model. The fine-tuning of the pre-trained model on 10 hours of Gujarati dataset significantly reduced WER from 55.8% to 32.4%. Arunkumar et al. (2022) has proposed ensemble model to combine the HuBERT and wav2vec2.0 to fine-tune ASR models. The performance of ensembled approach is compared with model generated from fine-tuning of indivisual HuBERT and wav2vec2.0 model. On the *LibriSpeech* dataset the ensemble model performed better.

Zheng et al. (2023) further explored the data selection approach for active learning. The authors extracted different level of granularity (K-means ID, phoneme, word) from unlabeled data using using HuBERT, wav2vec-U2.0 and WFST. They computed perplexity (PPL) and used PPL-based contrastive data selection approach. The proposed data selection framework effectively improved the word error rate (WER) by more than 11%.

The utilization of pseudo-labeling for unsupervised datasets has gained popularity as a means to mitigate the substantial effort and cost associated with generating labeled speech datasets. Xu et al. (2020) implemented iterative pseudo-labeling(IPL) technique for fine tuning the pre-trained model.This approach improved the word error rate on the *LibriSpeech* test dataset in both standard and low-resourced settings. Singh et al. (2023)

also experimented the pseudo-labels based self-training approach for low-resourced speech recognition. They fine tuned wav2vec2.0 based cross-lingual XLSR-53 model(Conneau et al., 2021) on the Punjavi speech dataset. The unlabeled dataset are labeled with pseudo-labels and marked with confidence score. Those data that pass the threshold confidence score are selected for the next level of fine-tuning. The process is repeated until the model stops improving.

Javed et al. (2022) has fine-tuned their proposed *IndicWav2Vec* model using Nepali OpenSLR dataset. When larger language model (LM_{large}), augmented lexicon, and re-scoring is used they improved the performance of model for Nepali speech input. As per Xu et al. (2021) the pseudo-labeling and self-training approach on *wav2vec2.0* can significantly improve the output in the low-resourced setting. The noisy student training with *SpecAugment* using giant Conformer models pre-trained using *wav2vec2.0* is experimented by Zhang et al. (2022). By doing this, they are able to achieve word-error-rates 1.4% on the LibriSpeech.

3 Methodology

3.1 Supervised Learning for Base model

Fine-tuning is a widely employed method in the field of machine learning. It involves the training of a pre-trained model, which has been previously trained on a substantial dataset for a task that is related, to be further trained on a more limited and specific dataset. The purpose of this additional training is to customize the model's performance for a particular task or domain.

We used Massively Multilingual Speech (MMS) - 1B (*mms-1b*) pre-trained model(Pratap et al., 2023) as the base model for the experiment. The model has been pre-trained using *wav2vec2.0*'s self-supervised training on a dataset of around 500,000 hours of speech data over more than 1,400 languages. The pre-trained model functions as a base model acquiring general speech patterns and representations from a wide range of audio data.

In the first phase (*Phase1 : Fine tuning the pre-trained model* as shown in the Figure 1) supervised approach is used for fine tuning the base model. This phase outputs the Fine Tuned (FT) model at level 0 (FT_{model_0}). The Labeled Dataset is used for the training.

3.2 Language Model

Language Models (LM) in ASR are essential for improving the accuracy and effectiveness of speech-to-text transcription. They provide language understanding, context, and the ability to adapt to different speakers and domains. The main reason of introducing LM is to evaluate the output of the ASR model in terms of the score while performing active learning. When a language model processes a sequence of words, it assigns a probability score to that sequence(LM_{Score}). The score represents how likely that particular sequence is, according to the learned patterns of the language model. In general, higher probability scores are assigned to more grammatically correct and contextually appropriate word sequences, whereas lower scores are assigned to less likely or incorrect sequences. This score is used as a dataset selection criteria in the active learning phase in both Algorithm 1 and 2.

KenLM Language Model Toolkit[1] is used for training the LM. We trained 5-gram KenLM LM using the text label of speech dataset(Kjartansson et al., 2018).

3.3 Active Learning

The speech corpus labeling task is expensive, time consuming, and resource-intensive. If we have seed data then we can use it to train the base model and use the trained model to further generate more data. This approach is known as Active learning approach. We used active learning approach on the *Phase 2* as presented in Figure 1. The FT_{model_0} is considered as the base model of this phase. The unlabeled dataset is labeled using the base model. Along with labeling, the LM_{Score} is also calculated. The LM_{Score} of each data in unlabeled dataset are compared with threshold value ($LM_{Threshold}$) to check if we can consider this data as semi-labeled data. If $LM_{Score} > LM_{Threshold}$ then this data is considered as semi-labeled data and it is merged with la-

[1]KenLM: https://kheafield.com/code/kenlm/

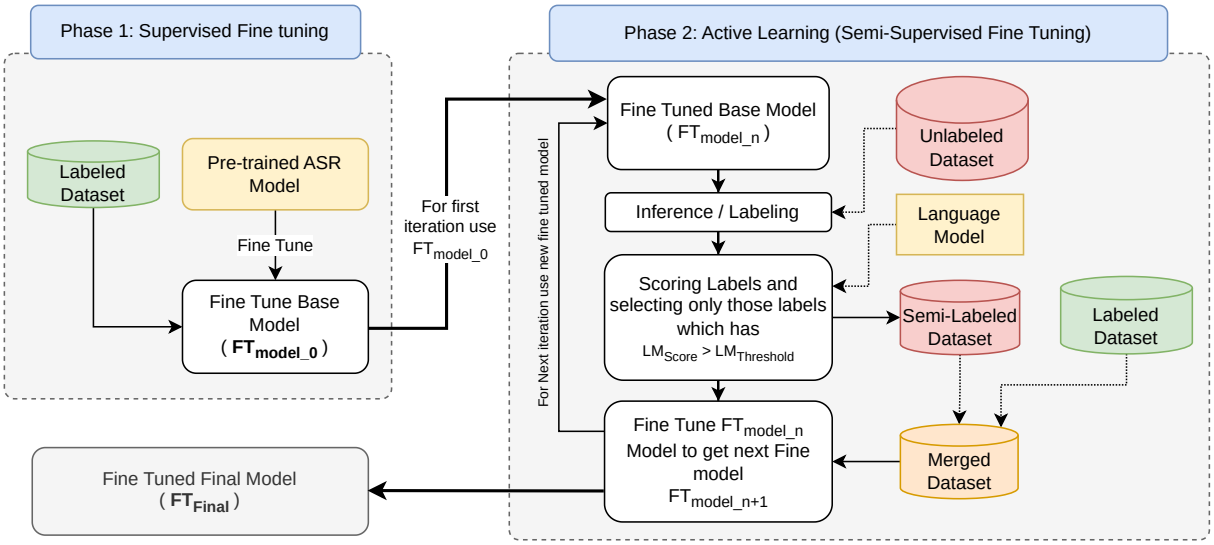

Figure 1: Architecture for semi-supervised iterative fine tuning of pre-trained model using Active Learning approach

beled dataset. The merged data is then used for further training.

Algorithm 1: Recursive fine tuning of the fine-tuned model

Input:
Pretrained Model: PT_{Model}
Labeled Train Dataset: $DS\_Train_{Labeled}$
Labeled Test Dataset: $DS\_Test_{Labeled}$
Unlabeled Dataset: $DS_{Unlabeled}$
Language Model: LM
Output: Final fine tuned model: FT_{Final}

FT_{model_0} = FineTune(PT_{Model}, $DS\_Train_{Labeled}$)
$FT_{final} = FT_{model\_0}$
do
  $DS_{semilabeled} = \{\ \}$
  foreach *speech in* $DS_{Unlabeled}$ **do**
    $label = FT_{final}$.Inference($speech$)
    $LM_{score} = LM$.Score($label$)
    if $LM_{score} > LM_{threshold}$ **then**
      $DS_{semilabeled}$.Append($speech$, $label$)
  $DS_{Merged} = DS_{semilabeled} \cup DS\_Train_{Labeled}$
  FT_{final} = FineTune(FT_{final}, DS_{Merged})
  CER = CalculateCER(FT_{final} , $DS\_Test_{Labeled}$)
while *(CER is improved)*
return FT_{Final}

The active learning phase (*Phase 2* as shown in Figure 1) is an iterative phase for fine tuning the base model. We followed two simple algorithms for the active learning. The Algorithm 1 is used for the recursive fine tuning of fine tuned model and the Algorithm 2 is used for the recursive fine tuning of the base model. The output of both algorithm are the fine-tuned ASR model FT_{Final}.

3.3.1 Requirements to perform active learning

The following inputs are required for active learning algorithms:

- Pretrained Model: PT_{Model} - It is (*mms-1b*) pre-trained model proposed by Pratap et al. (2023)
- Dataset: We need labeled as well as unlabeled datasets - 1) Labeled Train Dataset: $DS\_Train_{Labeled}$, 2) Labeled Test Dataset: $DS\_Test_{Labeled}$, and 3) Unlabeled Dataset: $DS_{Unlabeled}$
- Language Model: LM - It is 5-gram KenLM language model. The language model is used for calculating the score while generating the semi-labeled dataset.

3.3.2 Choosing $LM_{threshold}$

The convergence of the training toward minimal error is depends on the language models threshold score value $LM_{threshold}$. We performed training on smaller dataset to find the best fit among the set of the thresholds {-7, -6 , -5, -4, -3, -2, -1, 0, 1, 2, 3 ... }. Based on this training we selected -5 as a best fit for the given dataset. This selection criteria can be further improved.

3.3.3 Recursive fine tuning of the fine-tuned model

Algorithm 1 recursively fine-tunes the fine-tuned model. The FT_{model_0} is the base model that is used in the first iteration. The label is generated from the fine-tuned model and respective LM_{score} is calculated. If LM_{score} is within the accepted threshold (i.e. $LM_{threshold}$) then the data is considered as semi-labeled data and merged with $DS\_Train_{Labeled}$ dataset to form DS_{Merged}. The DS_{Merged} dataset are used for the next iteration. In this algorithm recently fine-tuned model is used for next iterations fine-tuning. The iteration continues until we get consistent CER.

3.3.4 Recursive fine tuning of the base model

Algorithm 2 recursively fine-tune the base model. The pre-trained mms-1b (PT_{Model}) model is first fine-tuned to build the FT_{Model_0}. This fine-tuned model is then considered as the base model for further fine-tuning. This algorithm is essentially the same as Algorithm 1, the differences is on *line:#11* where the recursive fine-tuning is done. In this algorithm we pass the FT_{Model_0} for further fine-tuning.

Algorithm 2: Recursive fine tuning of the base model

Input:
Pretrained Model: PT_{Model}
Labeled Train Dataset: $DS\_Train_{Labeled}$
Labeled Test Dataset: $DS\_Test_{Labeled}$
Unlabeled Dataset: $DS_{Unlabeled}$
Language Model: LM
Output: Final fine tuned model: FT_{Final}

1 FT_{model_0} = FineTune(PT_{Model}, $DS\_Train_{Labeled}$)
2 $FT_{final} = FT_{model\_0}$
3 **do**
4 $DS_{semilabeled} = \{\ \}$
5 **foreach** *speech in* $DS_{Unlabeled}$ **do**
6 $label = FT_{final}$.Inference(*speech*)
7 $LM_{score} = LM$.Score(*label*)
8 **if** $LM_{score} > LM_{threshold}$ **then**
9 $DS_{semilabeled}$.Append(*speech*, *label*)
10 $DS_{Merged} = DS_{semilabeled} \cup DS\_Train_{Labeled}$
11 FT_{final} = FineTune(FT_{model_0}, DS_{Merged})
12 CER = CalculateCER(FT_{final} , $DS\_Test_{Labeled}$)
13 **while** *(CER is improved)*
14 **return** FT_{Final}

4 Dataset

The validation of proposed model is conducted in a low-resourced environment. The Nepali language is used as the test language. Kjartansson et al. (2018) has developed a crowd-sourced speech corpus, known as OSLR53, specifically designed for Nepali and other languages that have limited resources available. The dataset can be accessed by the public through the Open SLR website[2]. The dataset consists of a total of 165 hours of recorded speech, which has been contributed by 527 distinct individuals. The dataset has a total of 157, 905 recorded phrases.

For the validation of our proposed model we splited OSLR53 dataset into three parts - 1) labeled train dataset ($DS\_Train_{Labeled}$), 2) labeled test dataset($DS\_Test_{Labeled}$), and 3) unlabeled dataset ($DS_{Unlabeled}$). The dataset details are summarized into the Table 1.

[2]SLR53 : https://openslr.org/54

Dataset	Details	
$DS\_Train_{Labeled}$	Duration	8.13 Hours
	Utterances	14474
$DS\_Test_{Labeled}$	Duration	1 Hour
	Utterances	1609
$DS_{Unlabeled}$	Duration	$\sim$ 80 Hours
	Utterances	$\sim$ 140K

Table 1: Training Dataset used for validating proposed model

Labeled Train/Test Dataset The labeled dataset contains the *tsv* file with *fileid*, *path* and *label*(transcribed text). Out of the total data, about $8Hours$ of data are considered as the training dataset whereas about $1Hour$ of data are considered as the test dataset. We used the test dataset to evaluate the fine-tuned ASR model.

Un-labeled Dataset Unlabeled dataset preparation task is comparatively easier than labeled dataset but still requires significant time to collect, curate, and pre-process. Unfortunately, the speech corpus for the unsupervised ASR task is not available. We are working on the unlabeled dataset preparation task. But at the moment we are not in a position to

use them. So for the validating our proposed model we have decided to use some portion of the data from OSLR53 as a unlabeled dataset after seprating train/test dataset.

5 Experiment, Result and Discussion

There are couple of pre-trained seed models (as explained in section 2) available for ASR task. These pre-trained models are trained on the larger speech corpus of single or multi-lingual speakers. For the experiment we used Massively Multilingual Speech (MMS) - 1B (*mms-1b*) as a pre-trained seed model.

The character based tokenizer is used to tokenize the label. The Devanagari characters that were used in the Nepali language are listed in Table 2 and are considered as vocabulary for fine-tuning. The vocabulary size used for training is 64.

Type	Symbols
Vowel	अ, आ, ई, इ, उ, ऊ, ए, ऐ, ऋ, ओ, औ
Consonants	क, ख, ग, घ, ङ, च, छ, ज,झ, ञ, ट, ठ, ड, ढ, ण, त, थ, द, ध,न, प, फ, ब, भ, म, य, र, ल, व, श, ष, स, ह
Vowel markers	ा, ि, ी, ु, ू, े, ै, ो, ौ
Other markers	् , ।, ं , ृ , "ँ etc.

Table 2: Devanagari characters used in Nepali Language

The fine-tuning, training, and executing the experiment was performed on a system equipped with an RTX 3090 GPU (24GB GPU Memory), a Ryzen 9 5600x CPU, and 32 GB of memory.

The Character Error Rate (CER) is used for the evaluation purpose. All the experiments are evaluated on same $DS\_Test_{Labeled}$ dataset.

We compare the performance of the base model and fine-tuned model. The base model is fine-tuned from *mms-1b* pre-trained model. The labeled dataset $DS\_Train_{Labeled}$ is used for fine-tuning. The evaluation is performed using $DS\_Test_{Labeled}$ dataset and resulted to 11% CER which is considered as baseline for further comparison. The experimental results are summarized in Table 3.

Experiment	Result (CER)
Base Model fine-tuned using $DS\_Train_{Labeled}$	11%
Active Learning using Algorithm 1 Recursive fine tuning of the fine-tuned model	**6.80%**
Active Learning using Algorithm 2 Recursive fine tuning of the base model	**6.77%**

Table 3: Result of various experiments

After successful execution of the Algorithm 1 (Recursive fine tuning of the fine-tuned model) and 2 (Recursive fine tuning of the base model) the returned fine-tuned model FT_{final} is evaluated on the $DS\_Test_{Labeled}$ Dataset with 6.80% and 6.77% CER respectively. Based on this, we can say that our fine-tuning approaches improve the performance significantly over the base model which is only trained using the labeled dataset.

The *mms-1b* is also fine-tuned by Pratap et al. (2023). Authors used *Dev* and *Test* Dataset for fine-tuning the base model. These datasets are not available in the public domain. The Common Crawl LM is used for decoding purpose. Their result for Nepali language in terms of CER is 8.3% amd 7.7% respectively on Dev and Test dataset. If we compare this work with ours, we can clearly say that our approach improves CER by 1% on the test dataset.

The supervised ASR models for the Nepali language proposed by Regmi and Bal (2021)(Joint CTC-Attention), Dhakal et al. (2022) (CNN-ResNet-BiLSTM), Banjara et al. (2020)(CNN-GRU) are state-of-art models which uses the OSLR53 dataset. Their reported CER are respectively, 0.30%, 17.07%, and 27.72%. Decreasing these CERs requires a significantly large labeled corpus. The effort for such corpus development is resource-intensive. This makes our approach the most

suitable approach for Nepali ASR as through this approach we can also achieve acceptable CER with comparatively a very small amount of resources.

6 Conclusion and Future Work

The Active learning-based self-supervised approach for developing ASR for low-resourced languages particularly Nepali is presented in this paper. The proposed method utilizes an iterative training approach, which has proven to be effective, especially in the context of languages with limited available resources. This iterative training process involves repeated cycles of learning and refining, allowing the system to adapt and improve its performance over time. By employing the suggested method, it is possible to eliminate the extensive labeling efforts in creating speech datasets for fully supervised ASR models. The experimental analysis demonstrates the effectiveness of our algorithms in terms of improving the performance of the ASR model in low-resourced settings of Nepali language. This research can be further extended for other languages which has limited language resources.

There are several areas for improvement in this research. We have considered only wav2vec2.0 based pre-trained seed model. We can test the algorithms on other pre-trained ASR models such as HuBERT. Language model is not being used yet for the decoding purpose. We can use the appropriate language model which is trained from larger text to improve the performance of the overall system. The language model threshold is fixed throughout the training. Some adaptive approach for the threshold calculation can be done.

References

A Arunkumar, Vrunda Nileshkumar Sukhadia, and Srinivasan Umesh. 2022. Investigation of Ensemble Features of Self-Supervised Pretrained Models for Automatic Speech Recognition. In *INTERSPEECH 2022*, pages 5145–5149. ISCA.

Alexei Baevski, Wei-Ning Hsu, Alexis Conneau, and Michael Auli. 2022. Unsupervised speech recognition. In *Advances in Neural Information Processing Systems 34*, arXiv:2105.11084, pages 27826–27839.

Alexei Baevski, Steffen Schneider, and Michael Auli. 2020a. Vq-Wav2vec: Self-Supervised Learning of Discrete Speech Representations. ArXiv preprint arXiv:1910.05453 (2020).

Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, and Michael Auli. 2020b. Wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations. In *Advances in neural information processing systems 33*, pages 12449–12460.

Janardan Banjara, Kaushal Raj Mishra, Jayshree Rathi, Karuna Karki, and Subarna Shakya. 2020. Nepali Speech Recognition using CNN and Sequence Models. In *2020 IEEE International Conference on Machine Learning and Applied Network Technologies (ICMLANT)*, pages 1–5. IEEE.

Yu-An Chung, Yu Zhang, Wei Han, Chung-Cheng Chiu, James Qin, Ruoming Pang, and Yonghui Wu. 2021. w2v-BERT: Combining Contrastive Learning and Masked Language Modeling for Self-Supervised Speech Pre-Training. In *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 244–250.

Alexis Conneau, Alexei Baevski, Ronan Collobert, Abdelrahman Mohamed, and Michael Auli. 2021. Unsupervised Cross-lingual Representation Learning for Speech Recognition. In *INTERSPEECH 2021*.

Manish Dhakal, Arman Chhetri, Aman Kumar Gupta, Prabin Lamichhane, Suraj Pandey, and Subarna Shakya. 2022. Automatic speech recognition for the Nepali language using CNN, bidirectional LSTM and ResNet. In *2022 International Conference on Inventive Computation Technologies (ICICT)*, pages 515–521.

Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. 2021. HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units. *IEEE/ACM Transactions on Audio, Speech, and Language Processing 29*, pages 3451–3460.

Tahir Javed, Sumanth Doddapaneni, Abhigyan Raman, Kaushal Santosh Bhogale, Gowtham Ramesh, Anoop Kunchukuttan, Pratyush Kumar, and Mitesh M. Khapra. 2022. Towards Building ASR Systems for the Next Billion Users. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 10813–10821.

Shreya Khare, Ashish Mittal, Anuj Diwan, Sunita Sarawagi, Preethi Jyothi, and Samarth Bharadwaj. 2021. Low Resource ASR: The Surprising Effectiveness of High Resource Transliteration. In *INTERSPEECH 2021*, pages 1529–1533. ISCA.

Oddur Kjartansson, Supheakmungkol Sarin, Knot Pipatsrisawat, Martin Jansche, and Linne Ha.

2018. Crowd-Sourced Speech Corpora for Javanese, Sundanese, Sinhala, Nepali, and Bangladeshi Bengali. In *6th Workshop on Spoken Language Technologies for Under-Resourced Languages (SLTU 2018)*, pages 52–55. ISCA.

Andy T. Liu, Shu-wen Yang, Po-Han Chi, Po-chun Hsu, and Hung-yi Lee. 2020. Mockingjay: Unsupervised speech representation learning with deep bidirectional transformer encoders. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6419–6423.

Da-Rong Liu, Kuan-Yu Chen, Hung-Yi Lee, and Lin-shan Lee. 2018. Completely Unsupervised Phoneme Recognition by Adversarially Learning Mapping Relationships from Audio Embeddings. In *INTERSPEECH 2018*, pages 3748–3752. ISCA.

Jian Luo, Jianzong Wang, Ning Cheng, and Jing Xiao. 2021. Loss Prediction: End-to-End Active Learning Approach For Speech Recognition. In *2021 International Joint Conference on Neural Networks (IJCNN). IEEE.*

Vineel Pratap, Andros Tjandra, Bowen Shi, Paden Tomasello Arun Babu, Sayani Kundu, Ali Elkahky, Zhaoheng Ni, Apoorv Vyas, Maryam Fazel-Zarandi Alexei Baevski, Yossi Adi, Xiaohui Zhang, Wei-Ning Hsu, Alexis Conneau, and Michael Auli. 2023. Scaling speech technology to 1,000+ languages. arXiv preprint arXiv:2305.13516 (2023).

Sunil Regmi and Bal Krishna Bal. 2021. An end-to-end speech recognition for the Nepali language. In *Proceedings of the 18th International Conference on Natural Language Processing (ICON)*, pages 180–185, National Institute of Technology Silchar, Silchar, India. NLP Association of India (NLPAI).

Steffen Schneider, Alexei Baevski, Ronan Collobert, and Michael Auli. 2019. Wav2vec: Unsupervised Pre-training for Speech Recognition. In *INTERSPEECH 2019*. ISCA.

Satwinder Singh, Feng Hou, and Ruili Wang. 2023. A Novel Self-training Approach for Low-resource Speech Recognition. In *NTERSPEECH 2023*, pages 1588–1592. ISCA.

Qiantong Xu, Alexei Baevski, Tatiana Likhomanenko, Paden Tomasello, Alexis Conneau, Ronan Collobert, Gabriel Synnaeve, and Michael Auli. 2021. Self-training and pre-training are complementary for speech recognition. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3030–3034.

Qiantong Xu, Tatiana Likhomanenko, Jacob Kahn, Awni Hannun, Gabriel Synnaeve, and Ronan Collobert. 2020. Iterative Pseudo-Labeling for Speech Recognition. arXiv preprint arXiv:2005.09267 (2020.

Yu Zhang, James Qin, Daniel S. Park, Wei Han, Chung-Cheng Chiu, Ruoming Pang, Quoc V. Le, and Yonghui Wu. 2022. Pushing the Limits of Semi-Supervised Learning for Automatic Speech Recognition. arXiv preprint arXiv:2010.10504 (2022).

Zhisheng Zheng, Ziyang Ma, Yu Wang, and Xie Chen. 2023. Unsupervised Active Learning: Optimizing Labeling Cost-Effectiveness for Automatic Speech Recognition. arXiv preprint arXiv:2308.14814 (2023).