KEC Conference 2026 DOI: 10.3126/injet.v3i2.95511

ISSN: 3021-940X (Print)

Self-Supervised and Semi-Supervised Learning for Nepali ASR with Limited Labeled Data

Rajesh Raskoti[1*], Kobid Karkee[2]

[1]*Department of Electronics and Computer Engineering, Himalaya College of Engineering, Kathmandu, Nepal, rajeshraskoti100@gmail.com*
[2]*Department of Electronics and Computer Engineering, Thapathali Campus, Kathmandu, Nepal, kobid@tcioe.edu.np*

Abstract

Nepali ASR suffers from a severe lack of manually transcribed data. This paper proposes a hybrid framework combining Self-Supervised Learning (SSL) and Semi-Supervised Learning (Semi-SL) to develop high-performance ASR systems under low-resource conditions. Three pre-trained transformer architectures, Wav2Vec2, XLSR-53, and CLSRIL-23 are adapted to the Nepali domain through a two-stage strategy: supervised CTC fine-tuning on limited labeled data, followed by an iterative pseudo-labeling loop that progressively incorporates unlabeled data. Experiments are conducted across four supervised training budgets (1h, 5h, 10h, 20h), evaluated on 16,136 test utterances from the SLR54 Nepali speech corpus. Results demonstrate that the proposed framework achieves competitive, with XLSR-53 reaching 23.05% WER and 5.05% CER at 20h, comparable to a strong supervised baseline. Critically, the proposed method shows measurable data-efficiency gains at intermediate training sizes, where for the XLSR-53 model at 20 hours, the proposed method reduces WER by 17.1% relative to the supervised baseline. Linguistic error analysis reveals that consonant confusion and vowel matra errors consistently account for 66 to 68% of all character-level errors, pointing to language-model integration as the highest-impact next step.

Keywords: Automatic Speech Recognition, Devanagari, Low-resource ASR, Nepali ASR, Pseudo-labeling, Self-Supervised Learning, Semi-Supervised Learning

1. Introduction

Automatic Speech Recognition (ASR) systems have become a critical asset for advancing human-computer interaction. However, conventional approaches predominantly rely on supervised deep learning, which requires massive amounts of manually transcribed data. This poses a major hurdle for languages like Nepali, where annotated datasets are extremely limited. The challenge is further amplified by the complex phonetic structure of the Devanagari script, which frequently causes standard models to achieve low accuracy. Self-supervised learning (SSL) has emerged as a powerful paradigm to address these challenges by learning robust acoustic representations from large quantities of unlabeled speech. Models such as Wav2Vec2 [1], XLSR-53 [2], and CLSRIL-23 [3] leverage contrastive objectives to pre-train on raw audio, enabling effective transfer to downstream ASR tasks with minimal labeled data. Semi-supervised self-training through pseudo-labeling further extends this capability by iteratively incorporating high-confidence unlabeled predictions into the training set [4]. This paper compares different SSL models for Nepali ASR. The work makes the following contributions: (1) a systematic evaluation of three SSL backbone architectures under identical conditions across four labeled data budgets (2) a semi-supervised self-training pipeline with confidence-based filtering and (3) a linguistic error analysis revealing Devanagari-specific failure modes that are largely architectural-invariant.

2. Related Work

Dhakal et al. [5] proposed a hybrid 1D-CNN + ResNet + BiLSTM model for Nepali ASR trained on the OpenSLR dataset, achieving a CER of 17.06%. Ghimire et al. [6] introduced a pronunciation-aware syllable tokenizer within a CNN-GRU architecture, reaching a CER of 8.09% and WER of 36.33%, demonstrating that language-specific tokenization is critical for morphologically complex scripts.

* *Corresponding author*

Poudel et al. [7] established a state-of-the-art benchmark with a Conformer-based system (NepConformer) achieving 6.01% CER and 23.96% WER using the SLR54 corpus. Ghimire et al. [8] demonstrated active learning-based self-training using a pre-trained mms-1b model, achieving 6.77% CER with only 8.13 hours of labeled data. Singh et al. [4] applied iterative pseudo-labeling on the Punjabi language using XLSR-53 with length-normalized confidence scores, achieving a 14.94% relative WER improvement.

The present work extends this line of research by providing a systematic multi-backbone comparison of SSL and semi-supervised methods for Nepali under controlled low-resource conditions, supplemented by Devanagari-specific linguistic error analysis.

3. Methodology

3.1. Dataset and Preprocessing

Experiments use the SLR54 Nepali Speech Corpus [9], comprising approximately 157,000 transcribed utterances from 527 speakers in .flac format. A speaker-disjoint split is applied: 10% for validation, 80% for training, and 10% for testing, ensuring that no speaker appears in more than one partition. From the training pool, hour-constrained labeled subsets of 1h, 5h, 10h, and 20h are sampled, while the remaining training audio is retained as the unlabeled pool. Audio preprocessing includes resampling to 16 kHz and voice activity detection (VAD) based silence trimming. Text preprocessing applies Unicode NFC normalization and removes non-Devanagari symbols. A character-level vocabulary is constructed from the training transcriptions.

3.2. SSL Backbone Architectures

Three pre-trained SSL models are evaluated: (1) Wav2Vec2 Base (facebook/wav2vec2-base), a 95M parameter model pre-trained on English LibriSpeech; (2) XLSR-53 300M (facebook/wav2vec2-XLSR-53-300m), a 300M parameter cross-lingual model pre-trained on 436,000 hours across 128 languages; and (3) CLSRIL-23 (nakyansh/wav2vec2-hindi-him-4200), a multilingual Indic model with closer acoustic proximity to Nepali. All models follow the wav2vec2 architecture: a convolutional feature encoder maps the raw waveform to latent representations, and a Transformer context network produces contextualized embedding optimized via a contrastive objective.

3.3. Training Tiers

Experiments are organized into three tiers: Baseline 1 (B1) uses randomly initialized models with supervised CTC fine-tuning on the labeled subset only, serving as the lower-bound reference. Baseline 2 (B2) starts from SSL pre-trained weights and applies CTC fine-tuning on the labeled subset, quantifying the SSL pre-training benefit. The Proposed System (PS) extends Baseline 2 with an iterative semi-supervised self-training loop.

3.4. Semi-Supervised Self-Training Loop

After initial CTC fine-tuning, the model decodes the unlabeled pool using beam search to generate pseudo-labels. A confidence score α is computed from the posterior distribution for each utterance. Only samples satisfying $\alpha \geq \tau$ are retained, where $\tau = 0.90$ is set empirically and drops by 0.15. Retained pseudo-labeled samples are merged with the original labeled set to form an augmented training corpus, and the model is retrained. This process is repeated for K=3 iterations. The training objective combines labeled and pseudo-labeled losses:

$$L^{total} = L^{labeled} + \lambda L^{pseudo} \quad \text{(Equation 1)}$$

where λ controls the contribution of pseudo-labeled data. The CTC loss enables alignment-free sequence-to-sequence training, particularly suited to low-resource settings.

3.5.Hyperparameters

We fine-tune our models using the AdamW optimizer, a learning rate of 3×10-5, and a weight decay of 0.01, starting with a 500-step warmup. Each model is trained for up to 30 epochs with an effective batch size of 32. To avoid overfitting, especially when data is limited, we freeze the convolutional feature encoder and use regularization techniques like attention dropout, hidden dropout, and layer drop, all set to 0.1. The best performing model is chosen based on validation WER, using early stopping with a patience of 5 checks and evaluating every 500 steps. In our semi-supervised approach, we run a self-training loop for three rounds, starting with a confidence threshold of 0.90, which drops by 0.15 each time, ending at 0.60. The initial threshold of 0.90 was chosen to ensure high precision of pseudo-labels when the model is uncertain. The decrease by 0.15 per iteration follows the observation that the model's confidence improves with each retraining round, allowing more but still reliable data to be included. The values were set based on preliminary experiments on a held-out validation set. Only pseudo-labels with confidence above these thresholds are used for retraining. The total training loss combines the supervised CTC loss on real labeled data and the CTC loss on pseudo-labeled data, weighted by a factor $\lambda = 0.5$ which equally weights labeled and pseudo-labeled losses. Lower values underutilize unlabeled data; higher values over-weight noisy pseudo-labels. This equal weighting is standard in self-training literature and yielded the best performance in our validation experiments at 10h and 20h.

3.6. System Diagram

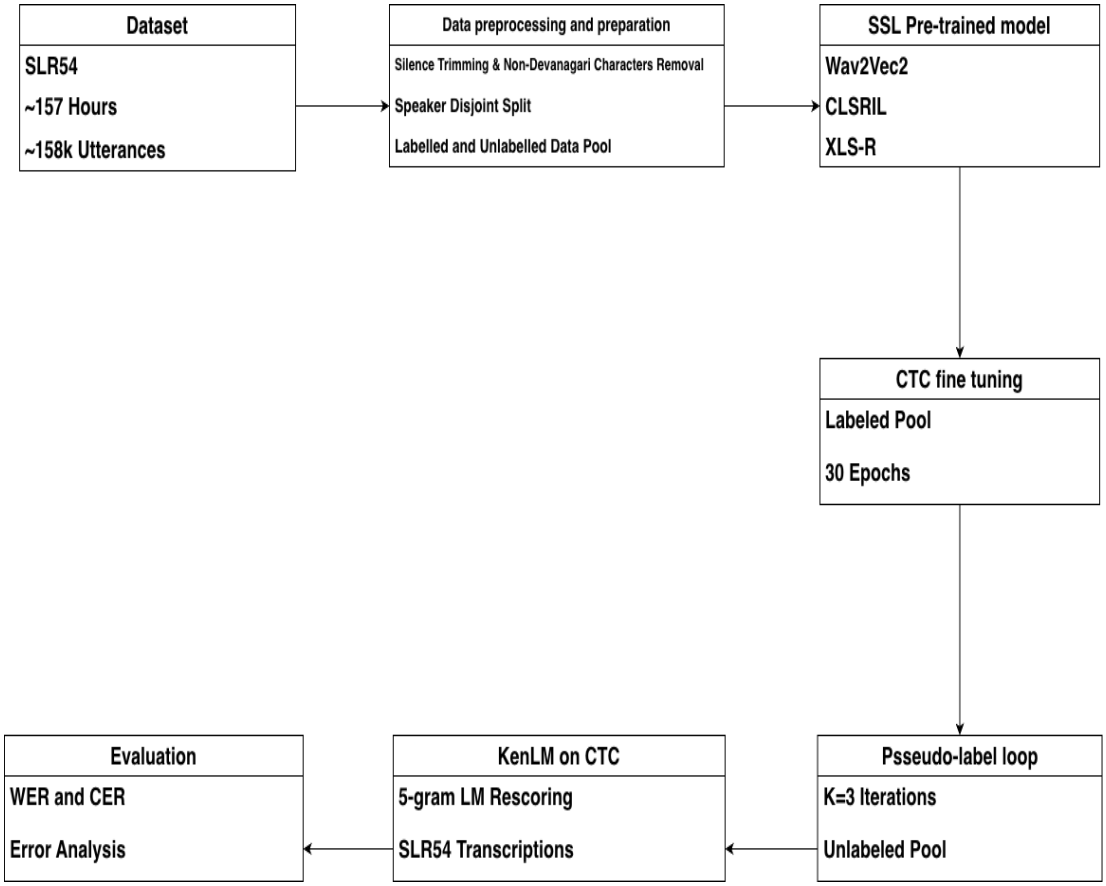

Figure 1. Block Diagram of the End-to-End Pipeline for Low-Resource Nepali ASR Using SSL Pre-training, Pseudo-label Self-Training

The diagram shows the main experimental setup for this work. First, the SLR54 Nepali speech dataset is divided so that speakers in the labeled and unlabeled groups don't overlap. The labeled group has subsets of 1, 5, 10, and 20 hours of data, while the remaining data form an unlabeled pool. Three different pre-trained speech models, Wav2Vec2, XLSR-53, and CLSRIL-23, are tested in three ways: (1) Baseline 1 is fune tuned on labeled data without any pre-training (2) Baseline 2 uses a pre-trained model that is then fine-tuned with labeled data (3) The Proposed system also adds the unlabeled data by generating and using

pseudo-labels in multiple rounds. All setups are evaluated using word error rate (WER) and character error rate (CER) on the same test set, allowing fair comparison of all models and methods.

3.7 Evaluation metrics

WER measures how many word-level errors a model makes compared to the reference transcription. It is calculated as:

$$WER = \frac{S + D + I}{N} \times 100\% \quad \text{(Equation 2)}$$

where S is substitutions, D is deletions, I is insertions, and N is the total number of words in the reference.
CER measures how many character-level errors a model makes compared to the reference transcription. It is calculated as:

$$CER = \frac{S_C + D_C + I_C}{N_C} \times 100\% \quad \textit{(Equation 3)}$$

where SC is character substitutions, DC is character deletions, IC is character insertions, and NC is the total number of characters in the reference.

4. Results and Discussion

4.1. Baseline 1: Supervised Only

Table 1 presents WER and CER for fully supervised training from random initialization. All models perform poorly, with WER exceeding 84% at 20h and frequently surpassing 100% at lower data budgets indicating more word-level insertions than reference words. Character-level representations are learned earlier than stable word boundaries, with CER dropping from ~78% at 1h to ~36% at 20h. XLSR-53 achieves the lowest B1 result at 20h (WER=84.64%, CER=32.78%). WER can go over 100% when a model adds way more words than are actually in the correct transcript. That's because extra words are counted in the total errors, but the total number of correct words stays the same, so the error rate can easily go above 100%.

Table 1. Baseline 1: WER (%) and CER (%) across models and data sizes.

Model	1h WER	1h CER	5h WER	5h CER	10h WER	10h CER	20h WER	20h CER
Wav2Vec2	107.36	78.13	100.00	73.85	98.84	51.29	89.25	35.96
XLSR-53	100.00	99.76	99.91	91.49	100.15	93.84	84.64	32.78
CLSRIL-23	104.76	76.77	100.25	91.97	98.27	50.97	90.57	36.95

4.2. Baseline 2: SSL Fine-Tuning

Table 2 shows substantial improvements with SSL pre-training. At 20h, Wav2Vec2 drops from 89.25% to 39.78% WER, XLSR-53 from 84.64% to 27.80%, and CLSRIL-23 from 90.57% to 25.66%. All three models still fail at 1h (WER=100%), suggesting a minimum data threshold below which fine-tuning cannot overcome the Devanagari script mismatch. CLSRIL-23 demonstrates the best data efficiency, reaching 40.63% WER at only 5h, a level Wav2Vec2 requires 20h to approach.

Table 2. Baseline 2: WER (%) and CER (%) across models and data sizes.

Model	1h WER	1h CER	5h WER	5h CER	10h WER	10h CER	20h WER	20h CER
Wav2Vec2	100.00	99.39	67.00	19.25	53.30	14.09	39.78	9.84
XLSR-53	100.00	100.00	49.66	12.66	36.70	8.76	27.80	6.32
CLSRIL-23	100.00	100.00	40.63	10.24	31.36	7.49	25.66	5.83

4.3. Proposed System: SSL + Pseudo-Labeling

Table 3 presents the proposed system results. XLSR-53 achieves its best result in the entire study at 20h (WER = 23.05%, CER=5.05%), outperforming its B2 counterpart by (17.1%, 20.1%) points, demonstrating a clear semi-supervised gain. CLSRIL-23 at 20h (WER = 25.59%, CER = 5.85%) remains on par with B2 (25.66%, 5.83%), suggesting saturation. Wav2Vec2 proposed at 20h worsens than baseline 2 because the model saturates with sufficient labeled data, and the pseudo‑labels (confidence threshold of 0.60) introduce more noise, a phenomenon known as pseudo‑label noise poisoning, which disproportionately affects models with weaker pre‑training. CLSRIL-23 outperforms XLSR-53 under purely supervised fine‑tuning because of its Indic phonological pre‑training. However, under the proposed semi‑supervised framework, XLSR-53 overtakes CLSRIL-23 due to its larger model capacity and more extensive pre‑training data, which produce higher‑quality pseudo‑labels that are amplified across iterations. This suggests that model capacity may be more important than linguistic proximity when applying iterative pseudo‑labeling to low‑resource languages.

Table 3. Proposed System: WER (%) and CER (%).

Model	1h WER	1h CER	5h WER	5h CER	10h WER	10h CER	20h WER	20h CER
Wav2Vec2	100.00	100.00	62.12	17.29	46.61	12.01	40.33	9.96
XLSR-53	100.00	100.00	42.50	10.43	30.37	6.98	23.05	5.05
CLSRIL-23	100.00	100.00	33.88	7.99	30.66	7.33	25.59	5.85

4.4. Learning Curves

Figure 2 shows WER vs. training data size for Baseline 2 (SSL fine-tuning). All three models start with very high error rates at the smallest data budget. As more labeled data is added, all models show steady improvement, with CLSRIL-23 consistently achieving the lowest errors across most budgets.

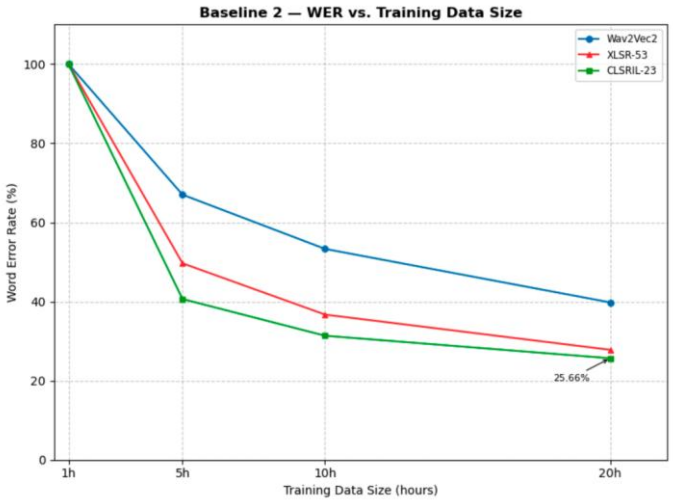

Figure 2. Baseline 2: WER vs. Training Data Size for all three SSL backbone models.

Figure 3 shows WER vs. training data size for proposed system. The overall shape is similar to Baseline 2, but a key difference is visible: at the smallest data budget, all models still perform poorly, but at the next budget, CLSRIL-23 drops sharply and matches or surpasses the other models. This indicates that the proposed method helps models adapt faster when very little labeled data is available.

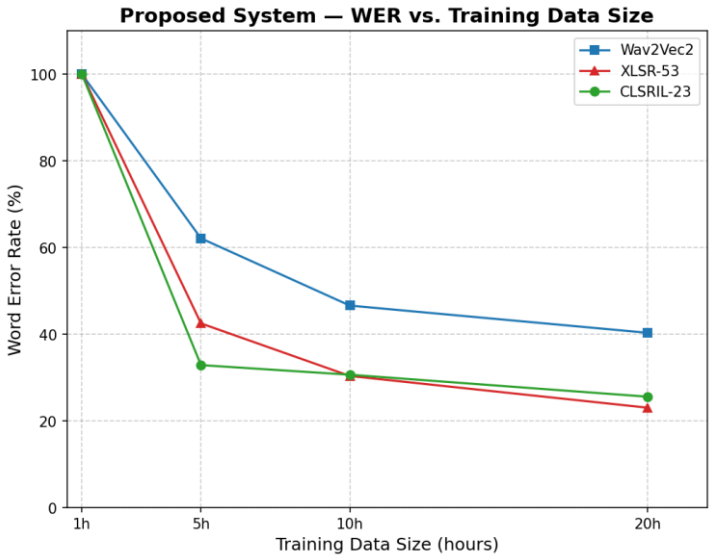

Figure 3. Proposed System: WER vs. Training Data Size for all three SSL backbone models.

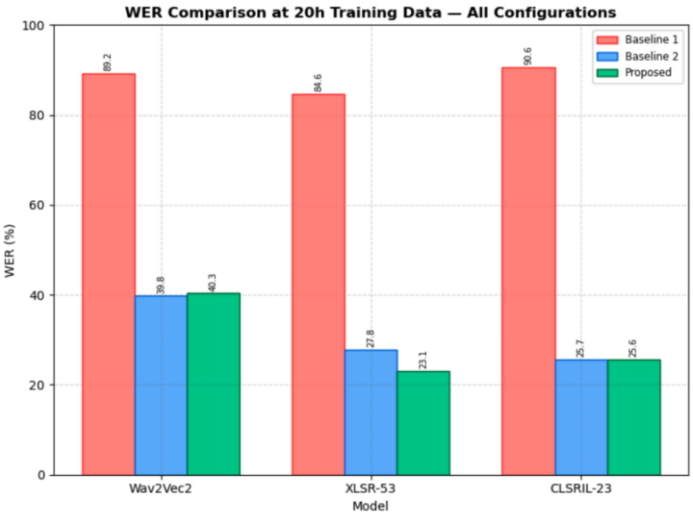

Figure 4. WER comparison at 20h across all model configurations (Baseline 1, Baseline 2, Proposed).

Figure 4 compares all three configurations at 20h. Across all models, SSL fine-tuning (B2) dramatically outperforms training from scratch (B1). The proposed system (PS) further improves XLSR-53 (23.05% vs 27.80% B2) but does not help CLSRIL-23 (25.59% vs 25.66% B2), confirming that CLSRIL-23 saturates earlier.

4.5. Linguistic Error Analysis

Table 4 presents per-category error breakdowns across selected best-performing models. The error distribution reveals a consistent and systematic pattern across all architectures and configurations. Consonant confusion (37-40%) is the dominant error category. Nepali's Devanagari script contains several phonetically similar consonant pairs, aspirated/unaspirated stops (ka/kha, ga/gha) and retroflex/dental

Table 4. Error category breakdown for selected best-performing models (16,136 utterance test set).

Model	Cons.	Vowel	Hal.	Del.	Ins.	Num.	Func.	Other
Wav2Vec2 B2 20h	13,991	9,887	5,359	2,162	1,064	1,023	976	608
XLSR-53 B2 20h	9,705	7,121	3,788	1,457	882	907	639	487
CLSRIL-23 B2 20h	8,975	6,882	3,531	1,343	1,005	892	736	461
XLSR-53 PS 10h	10,594	7,642	4,043	1,941	687	1,086	794	522
CLSRIL-23 PS 20h	8,933	6,892	3,532	1,497	845	935	713	457

contrasts (ta/Ta, da/Da), that are difficult to distinguish acoustically without a language model. Vowel matra errors (28-29%) constitute the second-largest category; Devanagari vowel diacritics are phonetically critical but acoustically subtle, especially in fast speech. Halanta errors (14-15%) arise from the conjunct consonant system, where the virama character suppresses the inherent vowel, a context-sensitivity not captured by the frame-wise CTC objective.

4.6. Comparison With Prior Work

Table 5. Comparison table of proposed work with prior work

System	Method	Data	WER (%)	CER (%)
Dhakal et. al. [5]	CNN+ResNet+BiLSTM	SLR54	-	17.06
Ghimire et. al.[6]	CNN-GRU + Syllabus Tokenizer	SLR54	36.33	8.09
Poudel et. al. [7]	Conformer	SLR54	23.96	6.01
Ours	**SSL + Pseudo-labeling**	**SLR54**	**23.05**	**5.05**

Table 5 compares the best result of the proposed framework against previously published Nepali ASR systems. Our XLSR-53 Proposed System achieves 23.05% WER and 5.05% CER using only 20 hours of labeled data, which is competitive with NepConformer (23.96% WER, 6.01% CER), a fully supervised Conformer trained on the complete SLR54 corpus.

5. Conclusion

In this paper, we thoroughly tested how well self-supervised and semi-supervised learning work for Nepali speech recognition when there isn't much labeled data. We compared three popular pre-trained models, Wav2Vec2, XLSR-53, and CLSRIL-23, using four different amounts of labeled data, all under the same experimental setup. Our results show that starting with a pre-trained model offers huge improvements over training from scratch, cutting word error rates by as much as 65%. Adding our iterative pseudo-labeling approach brought even more benefits for medium-sized datasets. For example, XLSR-53 reached a WER of 23.05% with 20 hours of labeled data, which is a 17.1% improvement over the standard supervised method. Looking at the types of mistakes made, we found that the models most often struggled with confusing consonants and vowel markings, no matter which model or training setup we used.

6.Limitations

Our study uses the SLR54 corpus, which contains read speech from 527 speakers. In real-world conditions with strong accents, background noise, or dialect variations (e.g., Eastern vs. Western Nepali), performance may degrade. The framework assumes that unlabeled audio is acoustically similar to the labeled data, if there is a large mismatch (e.g., telephone conversations vs. clean read speech), the quality of pseudo-labels will drop. Future work should evaluate on dialect-specific subsets to better understand where the framework breaks down.

7. Future Enhancement

The best next step is adding a language model because consonant confusion and vowel matra errors make up 66 to 68 % of all character‑level mistakes, and a language model can fix those using Nepali sound

patterns. The most practical immediate option is a simple n‑gram language model with beam search. Other important future work includes: (1) using a dynamic confidence threshold for picking pseudo‑labels instead of the fixed schedule we used, (2) adding more unlabeled Nepali audio from public broadcasts, and (3) testing newer models like Whisper and MMS to see if they still achieve the same data efficiency gains we found.

Acknowledgements

The authors gratefully acknowledge the Department of Electronics and Computer Engineering, Thapathali Campus, IOE, Tribhuvan University, for institutional support. The SLR54 Nepali Speech Corpus is gratefully acknowledged as the primary data resource for this research.

References

[1] Y. Zhou, A. Mohamed, and M. Auli A. Baevski, "wav2vec2: A framework for self-supervised learning of speech representations," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, pp. 12449-12460, 2020.

[2] A. Babu et al, "XLSR-53: Self-supervised cross-lingual speech representation learning at scale," in *Interspeech*, 2022, pp. 2278-2282.

[3] A. Gupta et al, "CLSRIL-23-23: Cross-lingual speech representations for Indic languages," in *arXiv preprint arXiv:2107.07459*, 2021.

[4] F. Hou, and R. Wang S. Singh, "A novel self-training approach for low-resource speech recognition," in *arXiv preprint arXiv:2308.05269*, 2023.

[5] M. Dhakal et al., "Automatic speech recognition for the Nepali language using CNN, bidirectional LSTM and ResNet," in *Int. Conf. Inventive Computation Technologies (ICICT)*, 2022, pp. 515-521.

[6] B. K. Bal, B. Prasain, and P. Poudyal R. R. Ghimire, "Pronunciation-aware syllable tokenizer for Nepali automatic speech recognition system," in *20th Int. Conf. Natural Language Processing (ICON)*, 2023, pp. 36–43.

[7] J. Poudel et al., "NepConformer: A conformer-based Nepali automatic speech recognition system," in *Int. Conf. Computing and Machine Learning*, 2025, pp. 167-178.

[8] B. K. Bal, and P. Poudyal R. R. Ghimire, "Active learning approach for fine-tuning pre-trained ASR model for a low-resourced language: A case study of Nepali," in *20th Int. Conf. Natural Language Processing (ICON)*, 2023, pp. 82–89.

[9] O. Kjartansson et al, "Crowd-sourced speech corpora for Javanese, Sundanese, Sinhala, Nepali, and Bangladeshi Bengali," in *6th Int. Workshop Spoken Language Technologies for Under-Resourced Languages (SLTU)*, 2018, pp. 52-55.