Adapter-Based Fine-Tuning for Low-Resource ASR

Domain Adaptation on Kinyarwanda Health Dataset using LoRA

Elvis Tata Tanghang,

getson2215@gmail.com,

November 2025

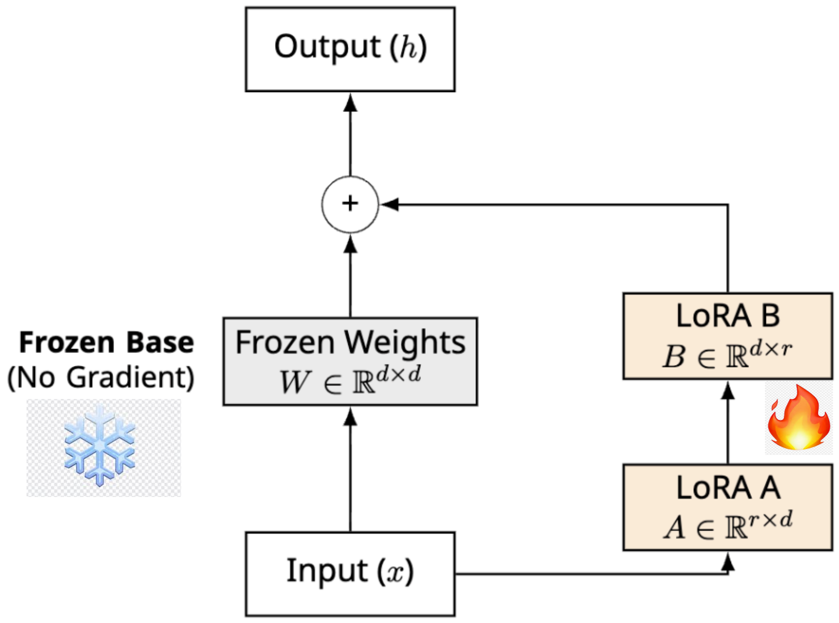
Abstract

This report details the implementation and results of the adapter-based-tuning strategy applied to a pretrained facebook/w2v-bert-2.0 (580M parameter + massive multilingual pre-training with 4.5M hours of unlabeled speech with a conformer architecture) as the base model. This model was fine-turned on Kinyarwanda Digital Umungada dataset provided during the ASR change to produce **badrex/w2v-bert-2.0-kinyarwanda-asr (**trained on 1000 hours of data covering several domains like government, finance and agriculture). Given that this model was trained (fine-tuned on Kinyarwanda) and has some basic knowledge on the language, we adapt this model to the health domain ensuring that it does not catastrophic lose its knowledge but carefully aligning it to the health domain. We train leveraging Low-Rank adaption (LoRA) which is a parameter efficient fineturning (PEFT) technique that requires training only **1.86%** of the base model params. We successfully demonstrated that we can avoid catastrophic forgetting using adapters by fine tuning the base model and improving its Word Error Rate(**WER**) from **6.51%** to **6.45%** showing an improvement of **0.06%.** We adapted a w2v-bert-2.0 speech encoder trained with CTC loss, inference with 2-gram language model with **11M** parameters to a 580M parameter base model on **NVIDIA A100-SXM4-80GB.** **Demo**

1. Methodology

1.1 Adapter Architecture. (Wav2Vec2-BERT + LoRA)

We propose a **Low-Rank Adaptation (LoRA)** approach applied to the **badrex/w2v-bert-2.0-kinyarwanda-asr** backbone by freezing the massive 580M parameter encoder model to preserve its feature extraction capabilities. We injected a LoRA matrix into the transformer layers and Feed Forward layer. This technique minimizes the risk of catastrophic forgetting by separating domain-specific knowledge from generalized knowledge. The target module was inserted to the all linear layers of the Transformer block: Attention (*WQ, WK, WV , WO)* and Feed-Forward (*Wup, Wdown*) as shown in Fig 1 below.


Output (h)
+
Frozen Base
(No Gradient)
Frozen Weights
$W \in \mathbb{R}^{d \times d}$
LoRA B
$B \in \mathbb{R}^{d \times r}$
LoRA A
$A \in \mathbb{R}^{r \times d}$
Input (x)

With r =16 and d = 32
Fig 1. Conceptual diagram of the LoRA integration. The frozen base model path preserves general knowledge, while the parallel LoRA path learns domain-specific features.

Setup	Parameters (Million)	Trainable %
Base Model	580 M	0.00
Adapted Module	11 M	100
Base + Adapter module	591 M	1.86

Table 1. Parameter of the base model and the added module

1.2 Training Strategy

The training pipeline focused on the Kinyarwanda Health dataset (afrivoice-kinyarwanda-health).

- **Data loading:** We used a custom torch.utils.data.Iterable dataset class to enable

  efficient, streaming-based reading of compressed .tar.xz audio archives. This avoids repeated disk I/O and large memory spikes.
- **Preprocessing:** Audio samples were resampled to 16kHz. Text labels were normalized to lowercase with special characters removed to match the model's vocabulary requirements.

- **Data Filtering:** A robust filter was implemented in the data loader to detect and skip audio samples where the waveform was entirely zero (silent or corrupt audio, amplitude $< 1e-5$), preventing training the model with noise
- **Text Normalization:** All training and evaluation transcripts were strictly cleaned to include only lowercase a-z, space, and the apostrophe ('), matching the reported successful preprocessing pipeline and preventing WER penalties from unpredicted characters.
- **Hyperparameters:**
  - **Learning Rate:** 5e−5
  - **Batch Size:** 1
  - **Gradient_accumulation_steps** = 4
  - **Epochs:** 1
  - **Max_step:** 4000
  - **Precision:** FP16 (Mixed Precision)

2. Experiments & Results

2.1 Evaluation Metric

We evaluated the base model and the fineturned model on the val_split of the data and measured the Word Error rate (WER) of the respective model. A common metric used in automatic speech recognition and Speech research. It measures how the transcripts are close to the target text. Another important metric is the CER character error rate which we also calculated but given that the instruction required only WER as a metric we reported on the WER only.

2.2 Results

Setup	**Greedy WER ⬇**	**WER + KenLM(2-gram) ⬇**	**KenLM improvement ⬆**
Base Model	6.51%	6.03%	**0.48%**
Adapted Model	**6.45%**	**5.98%**	0.47%
Reduction ⬆	**0.06%**	0.04%	0.02%

Table 2. Results of the experiment evaluation was done on val split of the dataset to calculate the Word Error Rate(WER)

The low parameter count of about 11M adapter model confirms the architectural efficiency. The reduction in WER from **6.51% to 6.45%** (0.06% improvement) validates that the adapters successfully learned subtle domain-specific phonetic patterns of the health dataset without altering the generalized acoustic features of the base model. While the improvement is small, the maintenance of high baseline performance (6.45% WER) with minimal added complexity highlights the stability and efficacy of the LoRA method for targeted domain adaptation. We also noticed that the KenLm improvement favored the base model more compared to the fineturned model. This is totally normal given the number of training parameters of the adapter compared to the base model. Also, the base model was trained on 1000h hours of kinyarwanda data so the model had learned basic understanding of the language and fine-tuning it for the health domain does not require significant learning, this explains why the performance improvement is minimal.

3. File Submission

- **Github link: https://github.com/tanettech/ASR_adapt/tree/main**
- Baseline checkpoints: https://huggingface.co/badrex/w2v-bert-2.0-kinyarwanda-asr
- Trained adapter weights: (github)
- Base_transcriptions.txt : (github)
- Finetune_transcription.txt: (github)
- Readme.md: (github)
-

4. Reproducibility Instructions

The solution is implemented in Python using PyTorch and Hugging Face Transformers.

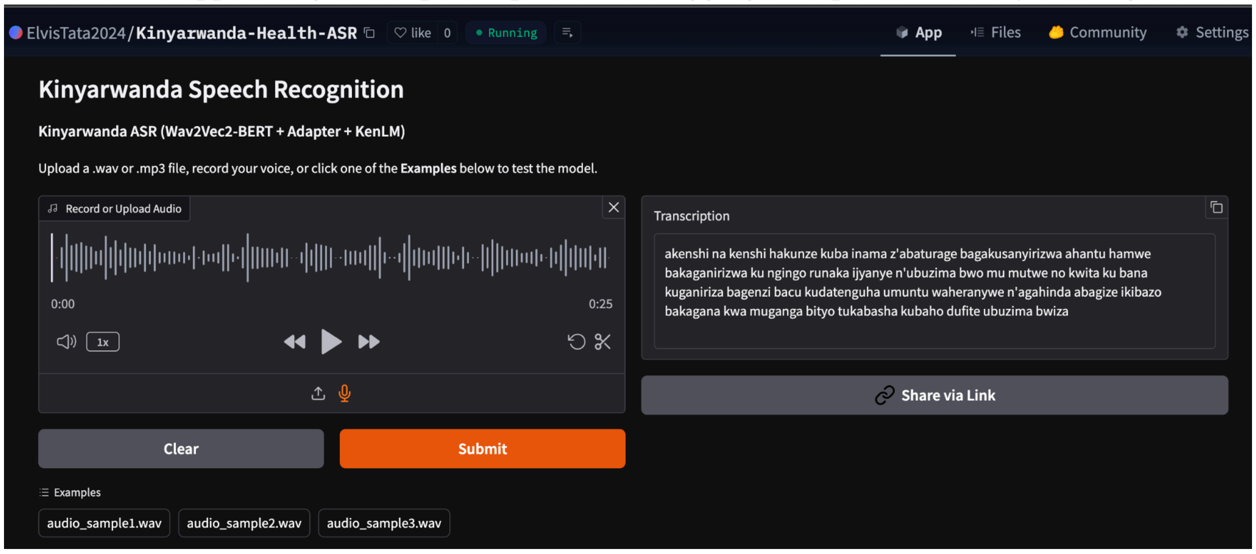
To reproduce the experimental setup and evaluation:

1. Environment Setup:
   Ensure Python 3.8+ and install dependencies:
   pip install transformers datasets torch librosa evaluate jiwer peft bitsandbytes

2. Data Preparation:
   Execute the data loading cells in Final_ASR_Adapt.ipynb. This will:
   - Download the afrivoice-kinyarwanda-health dataset.
   - Preprocess audio to 16kHz arrays.
   - Clean text labels.
3. Training:
   Run the training block. The script initializes the Wav2Vec2ForCTC model, attaches the LoRA config defined in LoraConfig, and executes the Trainer for 6 epochs.
   - *Checkpoint:* Models are saved in the ./wav2vec2-kinyarwanda-health directory.
4. **Evaluation:**
   Run the final evaluation cells to transcribe the test set.
   - The script uses evaluate.load("wer") to compute the metric.
   - Outputs transcriptions.txt and prints the final WER score.

5. Appendix

Gradio Demo app. (Merge checkpoint uploaded to huggingface space).Currently running



Kinyarwanda Speech Recognition

Kinyarwanda ASR (Wav2Vec2-BERT + Adapter + KenLM)

Upload a .wav or .mp3 file, record your voice, or click one of the **Examples** below to test the model.

Transcription

akenshi na kenshi hakunze kuba inama z'abaturage bagakusanyirizwa ahantu hamwe bakaganirizwa ku ngingo runaka ijyanye n'ubuzima bwo mu mutwe no kwita ku bana kuganiriza bagenzi bacu kudatenguha umuntu waheranywe n'agahinda abagize ikibazo bakagana kwa muganga bityo tukabasha kubaho dufite ubuzima bwiza

5. References

● B. Thomas, S. Kessler, and S. Karout, “Efficient Adapter Transfer of Self-Supervised Speech Models for Automatic Speech Recognition,” arXiv:2202.03218 (2022).

● W. Hou et al., “Exploiting Adapters for Cross-lingual Low-resource Speech Recognition,” arXiv:2105.11905 (2021).

● N. Houlsby et al., “Parameter-Efficient Transfer Learning for NLP,” arXiv:1902.00751 (2019).