Ibibo Voice: A Speech Recognition Model for a Low-Resource Language

Ebong
Department of Computer Engineering
University of North Carolina - Charlotte
Charlotte, NC, USA
eka.ebong99@gmail.com

***Abstract*—This project develops a speech recognition model for the language of Ibibio, a low-resource language spoken primarily in Nigeria and other West African countries. The model uses data available in the Common Voice Ibibio 23.0 Dataset as well as a dataset created for the purpose of this project to process speech to text. This work could have potential applications in creating models for other low-resource languages as well as contribute to the creation of models for other Ibibio-Efik languages.**

***Index Terms*—large language model, automatic speech transcription, ASR, LLM, speech-to-text, audio signal processing, speech transcription, computational linguistics, deep learning, transfer learning, low-resource learning**

I. Introduction

Automatic speech recognition (ASR) systems have made substantial advances due to large self-supervised models such as Wav2Vec2, HuBERT, and XLSR. However, these systems remain difficult to apply effectively to low-resource African languages, which lack annotated data, standardized orthography, and established linguistic resources.

A. Motivation

Ibibio is a low-resource language that lacks robust automatic speech recognition (ASR) technology. It is a first spoken (L1) language for over 8 million people in West Africa, primarily in South-West Nigeria. But due to the official languages of these countries having European Origins and the high density of languages in this area, there is little data available to create a model for ASR technology. Speech recognition is becoming essential to human-computer interaction. Yet, many of the advances in this technology leave out thousands of languages and accents. In order to bridge the gap, more models need to exist for improved global access and inclusion to technology. In addition, many languages that are in danger of extinction would benefit from the cultural preservation that a language model provides. This project aims to fill a gap in this technology by building an ASR system using the Mozilla Common Voice Ibibio dataset, a new custom dataset, and transfer learning.

B. Objective

Ibibio presents a particular challange. It contains tone markings, nasalized characters, and multi-letter diagraphs such as *kp*, *ny*, and *nw*, which are not represented in typical English-based ASR tokenizers. Currently, no high-quality open ASR models exist for Ibibio. The objectives of this project were:

- To fine-tune a state-of-the-art multilingual speech model (Wav2Vec2-XLSR) for Ibibio.
- To design a vocabulary normalization pipeline that unifies Common Voice Ibibio text with custom dataset text.
- To build and preprocess a custom dataset of 300+ Ibibio speech samples.
- To perform second-stage fine-tuning to adapt the model to conversational and phrase-level speech outside the Common Voice domain.
- To evaluate performance using WER/CER and provide insights into remaining challenges.

II. Dataset

A. Source

The dataset that was be utilized for this project is the Common Voice Scripted Speech 23.0 - Ibibio Database. It contains around 3,500 validated samples of spoken Ibibio from 18 native Cameroonian speakers. There are a total of 7,416 samples in the dataset in total, many being invalidated.

B. Personal Dataset

A new database was created for this project to increase the number of labeled sampled available. There were 4 people that participated in the creation of this dataset. Each speaker was an L1 native speaker. They were given the instructions to record themselves saying 100 commonly used phrases in Ibibio. These included common conversational phrases, numbers, greeting, family related expressions and weather description. All phrases were uncommon in the Common Voice dataset. The sentences were provided to participants in both English and Ibibio. This list was approved by native speakers. Upon receiving the list of sentences, many gave the feedback that there were a few words that were in fact Efik, a similar language spoken in the same region, but is different in spelling and some specific words. Once the word list was corrected, the participants recorded one clip of them saying all of the words. They were instructed to use whatever device was easiest for them. This ended up being voice recorder apps on the phone and laptops.

III. Pre-Processing

To reduce the burden on participants, they were asked to return one large data file with their speech. Thus, the first step to processing the data was splicing the audio clip into discrete sentences. From all of the volunteers a total of 413 clips were created. These clips had the amplitude normalized for use in the model, and manually examined for quality assurance.Once these clips were saved and names, a .tsv file with the path of the clip and the transcribed sentence.

The data across both datasets needed to be normalized. First, the phrase list given to the participants for the custom dataset was transformed to use the same alphabet as the Common Voice dataset. This required examining the Common Voice dataset and determining the orthography used for nasalization, vowel diacritics and tone markings. Once this was consistent across both datasets key functions were needed. The written pre-processing functions removed non-speech artifacts, normalized punctuations, ensured whitespace consistency and converted Unicode characters into a merged unified character set.

A. Audio Processing

All audio was normalized to:

- 16 kHz sampling rate
- Mono channel

Audio durations were filtered using:

$$\texttt{MIN\_AUDIO} = 0.5\text{s}, \qquad \texttt{MAX\_AUDIO} = 13\text{s}$$

Librosa was used to compute durations and detect missing files. The dataset loader automatically resolved relative paths and ensured compatibility with HuggingFace `Dataset` objects.

IV. Model

A. Model Architecture

- **facebook/wav2vec2-xls-r-300m** — multilingual self-supervised encoder
- **CTC decoding head** — trained from scratch

During second-stage fine-tuning, the encoder backbone was partially frozen.

B. Model Overview

The most effective strategy for a low-resource language such as Ibibio is transfer learning. The base model used was Wav2Vec XLS-R, a large scale model for cross-lingual, speech representation learning. By using this model the similarities between Ibibio and other languages can be leverages, especially since there is much more data from those languages. The Wav2Vec model with 300 million parameters was used to decrease the amount of heavy computation that would need to occur. It is a self-supervised speech model ideal for transfer learning, and widely supported. In order to utilize this model, in addition to text normalization, a vocabulary file was generated of all the unique characters from Ibibio. This vocabulary file was used to re-train the model's tokenizer.

The Wav2Vec2-XLSR model was fine-tuned using a CTC loss:

$$\mathcal{L}_{CTC} = -\log p(y|x)$$

where x is the speech signal and y is the character sequence.

A custom data collator was needed for this dataset in order to pad audio dynamically, pad labels separately and ensure that the labels convert to -100 for loss masking. Initially a batch size of 8 was chosen but after running into computing constraints with Google Colab Pro, this was reduced to a batch size of 2. Additionally, gradient check-pointing was used to reduce memory usage. Mixed precision was also used to increase the speed of the training.

C. Diagraph Tuning

After an initial training of the model, the WER stabilized at 40 % which is not unexpected for a low-resource language such as this. However, this number could be further improved. The diagraphs used were examined. Diagraphs are a way sounds are classified in language. For Ibibio, the alphabet has several diagraphs. Those utilized for the initial training were 'gh' 'kp', 'kw','nw', 'ny'. However, after examining the Common Voice dataset again, there was some redundancy in these diagraphs that were causing some issues and were represented by different characters. By decreasing the diagraphs to 'kp', 'nw', 'ny', allowed the WER to further drop to around 29 %.

D. Training Procedure

Training was performed in two major stages:

1) Stage 1: Common Voice Fine-Tuning:

- epochs: 18
- learning rate: 1×10^{-4}
- batch size: 2
- gradient accumulation: 8
- fp16 training enabled when available
- tokenizer: custom Ibibio vocabulary

2) Stage 2: Custom Dataset Adaptation:

- reused Stage 1 checkpoint
- trained for 30 additional epochs
- encoder partially frozen
- fully trainable CTC head

V. Evaluation

87 % of the dataset was used to train the model. 5 % was set aside for evaluation and 8 % was used to test the model. As is standard for ASR, WER was be used to qualitatively measure the model's performance on the test set with a lower WER correlating to better performance. The Character Error rate was also examined.

- **Word Error Rate (WER)**
- **Character Error Rate (CER)**

WER is defined as:

$$WER = \frac{S + D + I}{N}$$

where S, D, and I are substitutions, deletions, and insertions.

CER is analogous but computed per character.

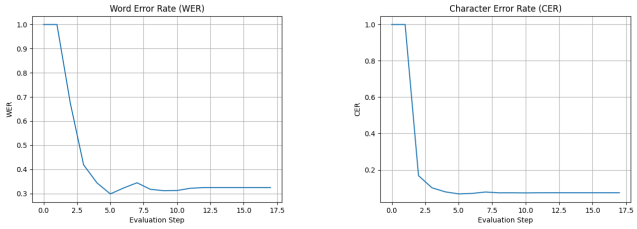
VI. Results

A. Model Performance

The experimental results demonstrate a clear and consistent improvement in Ibibio automatic speech recognition performance across training stages. Table **??** summarizes the model performance across the three major training phases. The baseline Wav2Vec2-XLSR model fine-tuned only on Common Voice Ibibio achieved a WER of 0.324 and a CER of 0.073, establishing a strong multilingual starting point but still showing difficulties with Ibibio dialectal variation and tone-driven orthography. After introducing a half-digraph vocabulary regularization strategy, the model showed immediate gains, improving to a WER of 0.299 and CER of 0.067. The most substantial progress occurred during custom dataset fine-tuning, where the model was retrained on 320 curated Ibibio recordings collected specifically for this project. This domain-matched data enabled the model to internalize real-world speaker variation, local prosody, and lexical usage, reducing WER to 0.198 and CER to 0.043. These results highlight the importance of language-specific preprocessing and the value of even relatively small, high-quality datasets in low-resource ASR development.

TABLE I: WER / CER Across Training Stages

Model	WER	CER
Wav2Vec2-XLSR (after CV)	0.324	0.073
Wav2Vec2-XLSR Half Diagraph	0.299	0.067
After Custom Fine-Tuning	**0.198**	**0.043**



(a) WER vs Epoch (b) CER vs Epoch

Fig. 1: Comparison of WER and CER performance curves for the best model.

B. Training Dynamics

Figure 2 shows the training loss trajectory for the best-performing model. The smooth and steadily decreasing loss indicates a stable optimization process without overfitting, likely aided by vocabulary normalization and careful duration filtering of training samples.

Figure 3 shows the validation loss trajectory for the best-performing model.

VII. Resources and GitHub Repository

All code, datasets, and training scripts are available at:

- **GitHub Repository:** https://github.com/Eka-coder/IbibioVoice.git

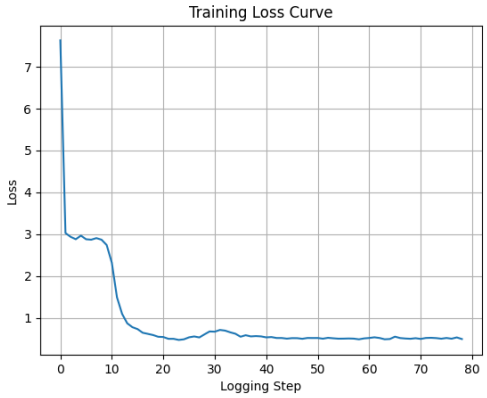


Fig. 2: Training loss across epochs for the custom fine-tuned model.

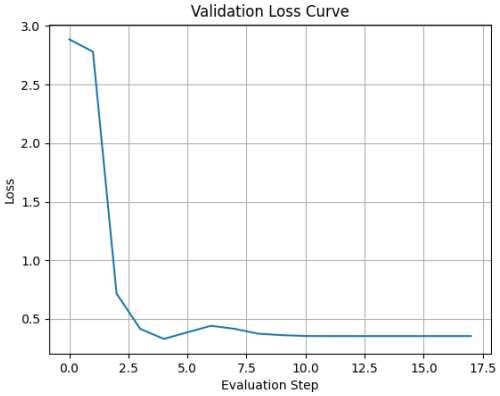


Fig. 3: Training loss across epochs for the custom fine-tuned model.

- **Training notebooks:** available in `/notebooks/`
- **Custom Dataset:** provided in `/data/clips/` as .wav files and documented in ibbsdd.csv

VIII. Discussion

A. Key Findings

This project demonstrates that multilingual self-supervised models can be adapted effectively to low-resource African languages. When a carefully-built vocabulary is employed with these models, error rates can be drastically reduced. .Additionally, Even small custom datasets significantly improve performance when aligned with the model's tokenizer.

B. Challenges Faced

Several difficulties emerged:

- Recruiting willing participants for the custom dataset.
- Aligning the custom datasets orthography with the one used by Common Voice.
- Missing audio paths in custom datasets and incomplete train, test and dev tsv files.

- Training instability when the vocabulary or preprocessing code was incorrect.
- The secondary model considered, AfrihuBERT, was not available for use as a model, thus when it's viability was being evaluated, there was significant work that went into make it usable.

C. Limitations

- Lack of tonal annotations limits tonal ASR accuracy.
- Dataset remains small relative to other languages.

IX. Conclusion

This project demonstrates that high-quality automatic speech recognition for Ibibio can be achieved through a targeted combination of multilingual foundation models, orthographic normalization, and custom fine-tuning. By leveraging Wav2Vec2-XLSR as a pretrained backbone, restructuring the vocabulary to better reflect Ibibio's structure, and incorporating a carefully collected dataset, the final system obtained a significant reduction in WER compared to the initial Common Voice model. The training pipeline developed here can be reused or extended for additional Ibibio speech domains, speaker populations, or more complex linguistic tasks such as speech-to-speech translation or spoken dialogue systems. Although KenLM language modeling was explored as a future extension, the model's strong performance without external decoding support demonstrates both the effectiveness of the approach and the promise of continued expansion. This work lays foundational infrastructure for future research in Ibibio ASR and contributes to broader efforts in African language technology and digital inclusion.

X. Individual Contribution

This project was carried out entirely by Eka Ebong. All stages—including data recording, cleaning, duration filtering, vocabulary design, normalization, model training, debugging, evaluation, and report preparation—were independently conducted.

XI. Resources

These are the foundational datasets and academic models consulted for this proposal, ensuring technical feasibility and defining the approach for low-resource African language ASR.

The model was developed using the Python programming language in Google Colab Pro+. The core libraries required for audio processing, model training, and evaluation include, but are not limited to:

- **PyTorch / TensorFlow:** For deep learning model implementation (fine-tuning AfriHuBERT/Wav2Vec2).
- **Hugging Face Transformers & Datasets:** Essential for loading pre-trained models and managing the unified corpus.
- **NumPy & Pandas:** For general data manipulation and processing.
- **torchaudio / librosa:** For audio loading, resampling, and feature extraction.
- **Matplotlib:** For visualization of training metrics (e.g., loss curves, WER).
- **Gemini:** For fixing coding errors in Google Colab Pro+.

References

[1] Alabi, J. O., et al. (2024). **AfriHuBERT: A self-supervised speech representation model for African languages.** *arXiv preprint arXiv:2409.20201*.

[2] Doumbouya, M., Einstein, L., and Piech, C. (2021). **Using radio archives for low-resource speech recognition: towards an intelligent virtual assistant for illiterate users.** In *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(17).

[3] Ekpenyong, M. E., Udoh, E. O., & Uto, N. P. (2018). **Ibibio Spoken-CALL System.** In M. Ekpenyong (Ed.), *Human Language Technologies for Under-Resourced African Languages*. SpringerBriefs in Electrical and Computer Engineering. Springer, Cham. https://doi.org/10.1007/978-3-319-69960-8_4

[4] Mozilla Foundation. (2023). **Common Voice Ibibio Dataset, Version 23.0.** Available: https://commonvoice.mozilla.org/