Application of the multilingual acoustic representation model XLSR-53 for the transcription of Ewondo

Nzeuhang Yannick Yomie, Yonta Paulin Melatagia, Lecouteux Benjamin

Application of the multilingual acoustic representation model XLSR-53 for the transcription of Ewondo

Yannick Yomie Nzeuhang[1], **Paulin Melatagia Yonta**[*1,2], **Benjamin Lecouteux**[3]

[1]Department of Computer Sciences, University of Yaounde I, Cameroon
[2]IRD, UMMISCO, F-93143, Bondy, France
[3]Univ. Grenoble Alpes, CNRS, Grenoble INP, LIG, 38000 Grenoble, France

*E-mail : Yannick.Yomie yynzeuhang@gmail.com

Abstract

Recently popularized self-supervised models appear as a solution to the problem of low data availability via parsimonious learning transfer. We investigate the effectiveness of these multilingual acoustic models, in this case wav2vec 2.0 XLSR-53, for the transcription task of the Ewondo language (spoken in Cameroon). The experiments were conducted on 24 minutes of speech constructed from 103 read sentences. Despite a strong generalization capacity of multilingual acoustic model, preliminary results show that the distance between XLSR-53 embedded languages (English, French, Spanish, German, Mandarin, . . .) and Ewondo strongly impacts the performance of the transcription model. The highest performances obtained are around 70.8% on the WER and 28% on the CER. An analysis of these preliminary results is carried out and then interpreted; in order to ultimately propose effective ways of improvement.

Keywords

Low resource language; Self-supervised model ; XLSR-53 ; Transcription ; Ewondo

I INTRODUCTION

Self-supervised learning is a deep learning method for learning robust representations from unlabeled data. the main idea is to automatically generate labels for a simple pretext task, enabling the model to better understand the given structure, and then to use this learned information for a more complex target task. This method has recently been widely illustrated in speech processing, notably by the multilingual acoustic model **wav2vec 2.0 XLSR-53** [7], which delivers impressive results for automatic speech recognition (ASR) tasks, even on small datasets. By these fact this model presents itself as a solution for low ressources languages for which automatic speech processing tasks are difficult to address by deep learning, due to the difficulty of building a large dataset. Ewondo, language from central Cameroon falls into this category of language.

Our aim is to evaluate effectiveness of multilingual acoustic model on Ewondo, which has the particularity of being tonal. To achieve this goal, we have built several ASR models based on *wav2vec 2.0*[6] in various configurations, we have evaluated the performance on word error rate (WER) and character error rate (CER). To the best of our knowledge, this study is the first of its kind for the Ewondo language. Our contribution in this paper is twofold: 1) The construction

of a basic speech recognition model in Ewondo 2) Preliminary performance evaluation of a multilingual acoustic model for Ewondo, which allows us to outline paths for the construction of a robust model.

The rest of the paper is structured as follows. In section 2 we briefly introduce and discuss the background related to this work. Section 3 presents our approach. In section 4 we describe the experiments and discuss the results. Finally, we conclude in section 4.

II BACKGROUND

2.1 Ewondo language

Ewondo is a bantue language of central Cameroon, it is spoken by the Ewondo people in Cameroon, predominantly in the central and southern regions. It derived from the Fang-Beti language, which belong to the extensive Bantu language family, known for its diversity and widespread presence across sub-Saharan Africa.

The linguistic and cultural landscape of Ewondo is deeply rooted in the traditions and heritage of the Ewondo people. This language serves as a vital means of communication within the community, reflecting the rich history and social intricacies of its speakers. With its prevalence in urban areas, particularly in the capital city, Yaoundé, Ewondo plays a crucial role in daily interactions, commerce, and cultural expression.

The phonetics of Ewondo involve a set of distinctive consonants and vowels, contributing to its unique sound system. Pronunciation nuances, intonation patterns, and rhythmic elements are integral to conveying meaning accurately in spoken Ewondo. The language also incorporates a range of tones, a common feature in many Bantu languages, which further adds depth and complexity to its oral expression. In fact Ewondo is a tonal language, meaning that word meanings differ according to pitch, even if the consonants and vowels are the same [1] (Table 2 shows pairs of words of this type). The Ewondo language has 8 tones (Table 1), divided into punctual tones, which are tones for which the pitch remains invariable from the beginning to the end of the pronunciation, and modular tones, which vary in pitch.

Efforts to document and preserve Ewondo, both in written and oral forms, contribute to safeguarding the linguistic diversity of Cameroon. Like all Cameroonian languages, Ewondo uses the GACL[1] alphabet (general alphabet of Cameroonian languages) based on the Latin alphabet. As with many endanged languages, Ewondo faces challenges such as globalization, urbanization, and the dominance of major languages. However, initiatives to promote language education, cultural exchange, and community engagement are crucial for the continued vitality of Ewondo and its significance in the mosaic of Cameroon's linguistic heritage; this work is also in line with this aim. Despite of efforts and like all the languages of Cameroon, Ewondo remains a low ressource language, i.e. numerical resources are almost non-existent. This constitutes a major difficulty for deep learning approaches to solving tasks such as speech recognition. However, recent approaches based on self-supervised models make it possible to tackle this type of language.

[1] https://www.silcam.org/fr/resources/archives/32295

Table 1: Tones in Ewondo language

Pontuel tone		Modular tone	
Denomination	Notation	Denomination	Notation
Low[Tb]	[51v, v́]		
High[HT]	[44v, v́]	High-Low[HLT]	[51v, v̂]
Medium[MT]	[33v, v̄]		
M-Low [MLT]	[|v]		
Supra-High[SHT]	[55v]	Low-High[LHT]	[15v, v̌]
Infra-LOW[SIL]	[12v]		

Table 2: Words that differ only in tone

Words	Translation	Words	Transl
minkud	*bag*	*minkúd*	*clo*
zám	*raffia*	*zàm*	*good*
bám	*to scold*	*bam*	*to w*
bóg	*to pil up*	*bog*	*to ex*
tag	*to rejoice*	*tág*	*to cla*

2.2 ASR with Self-supervised model

Self-supervised learning is a machine learning paradigm where a model learns to make predictions about certain aspects of the input data without explicit supervision from labeled examples. First the NLP (Natural language processing) plume this approach has gained popularity for its ability to utilize vast amounts of unlabeled data, often abundant in real-world scenarios. In fact, in self-supervised learning, the learning algorithm creates its own supervision signal through a carefully designed pretext task. The pretext task is a task that is generated from the input data itself and doesn't require external annotations. As is shown in Figure 1 the model is trained to solve this pretext task, and the acquired knowledge can then be transferred to downstream tasks where labeled data might be scarce.

The literature is replete with a number of self-supervised acoustic models (for the review of these model the reader can refers to [13]), but we have chosen to exploit the XLSR-53 a crosslingual version of wav2vec 2.0 [6] for its promising results on languages with small amounts of data. This model uses a pre-training task similar to BERT[4], illustrated in Fig 2. This pre-training task consists of randomly masking words in sentences and asking the model to find the correct words. In the case of speech, parts of the signal are masked.

III WAV2VEC2.0 FOR EWONDO

We can subdivide our model in two parts; the cross-lingual speech representations (XLSR-53) [7] as a feature extractor and connectionist temporal classifier(CTC) [2] as a classifier. This section present the overall design of the model and different configurations used during experiments.

3.1 Model

Our work is based on the Wav2Vec 2.0 [2] model. Overall, the Wav2vec 2.0 uses an auxiliary task similar to BERT [13], where certain parts of the signal are masked in order to be reconstructed by the system, it is trained by predicting speech units for masked parts of the audio. As shown in Figure 4 we use as feature extractor the cross-lingual speech representations (XLSR-53)[7] version which is a multilingual representation model pre-trained on 53 languages. The multilingualism of the model increases its generalization capabilities, and this need for generalization is further exacerbated in our context where the small amount of data forces us to freeze

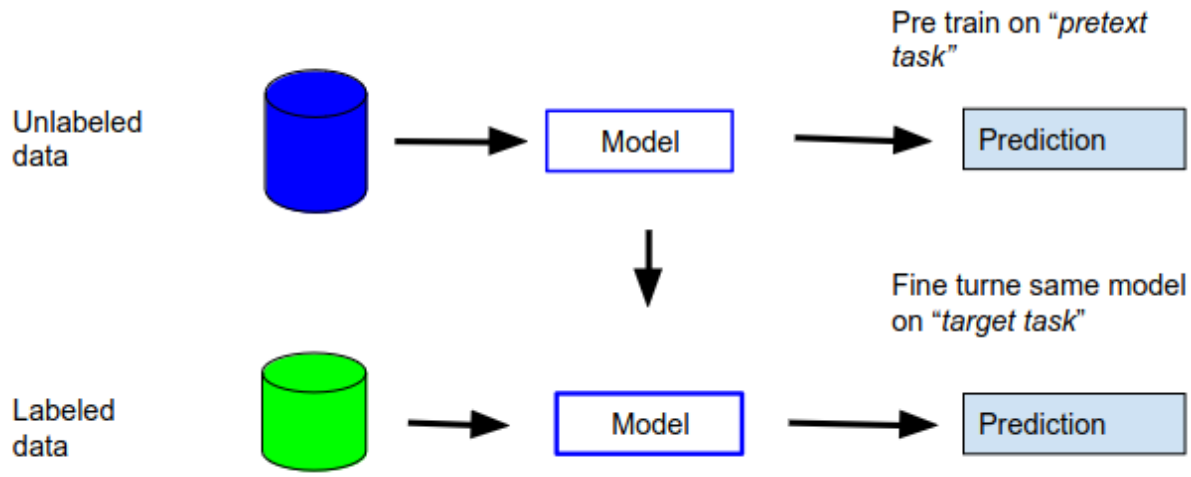

Figure 1: Principe of self-supervised learning.

the weights of the extraction model, i.e. the weights of XLSR-53 will not be modified during training. Our model is an encoder-decoder architecture where XLSR-53 acts like a encoder so it produce the latent representation of speech which is use by a decoder, connectionist temporal classifier (CTC) in this case. The CTC decoder model is a simple linear transformation followed by a softmax normalization. This layer should project output vector of encoder into the dimensionality of the output alphabet for each position in the output sequence. The main feature of this decoder is that it does not require strict alignment between the audio signal and its transcription, i.e. it only needs the input vectors (produced by XLSR-53) and the overall output sentence for training rather than a strict correspondence between input vector segments and output sentence segments. Let's take a closer look at the formal description of the fonctionnement of each part of our model.

IV WAV2VEC2.0 FOR EWONDO

We can subdivide our model in two parts; the cross-lingual speech representations (XLSR-53) [7] as a feature extractor and connectionist temporal classifier(CTC) [2] as a classifier. This section present the overall design of the model and different configurations used during experiments.

4.1 Model

Our work is based on the Wav2Vec 2.0 [2] model. Overall, the Wav2vec 2.0 uses an auxiliary task similar to BERT [13], where certain parts of the signal are masked in order to be reconstructed by the system, it is trained by predicting speech units for masked parts of the audio. As shown in Figure 4 we use as feature extractor the cross-lingual speech representations (XLSR-53)[7] version which is a multilingual representation model pre-trained on 53 languages. The multilingualism of the model increases its generalization capabilities, and this need for generalization is further exacerbated in our context where the small amount of data forces us to freeze the weights of the extraction model, i.e. the weights of XLSR-53 will not be modified during training. Our model is an encoder-decoder architecture where XLSR-53 acts like a encoder so it produce the latent representation of speech which is use by a decoder, connectionist temporal classifier (CTC) in this case. The CTC decoder model is a simple linear transformation followed by a softmax normalization. This layer should project output vector of encoder into the

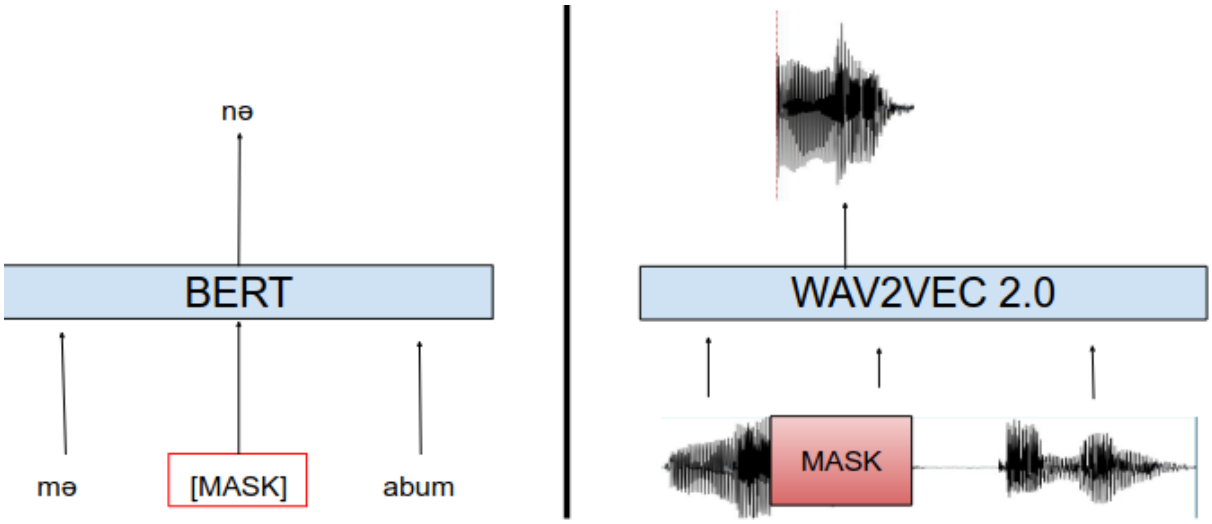

Figure 2: **Left**. Example of a BERT pre-training task with the sentence "m n abum" (I'm pregnant), the word "n" is hidden and the model must predict it. **Rigth**. The same concept is applied to the audio signal, where certain portions are masked and wav2vec 2.0 must predict them.

dimensionality of the output alphabet for each position in the output sequence. The main feature of this decoder is that it does not require strict alignment between the audio signal and its transcription, i.e. it only needs the input vectors (produced by XLSR-53) and the overall output sentence for training rather than a strict correspondence between input vector segments and output sentence segments. Let's take a closer look at the formal description of the fonctionnement of each part of our model.

Encoder. This XLSR-53 is a multilingual version of wav2vec 2.0 that consists of three parts: firstly, the feature encoder, which contains a multilayer convolutional neural network to process the raw waveform of audio speech. Secondly, the transformers, which are fed by the encoded feature and learn a contextualized representation from it, and thirdly the quantization module for selecting the speech unit to be learned from the latent representation space produced by the feature encoder. The purpose of this third part is to reduce the cardinality of the representation space and can be thought of as a function q that maps any vector x in the latent space to a vector $q(x)$ in a small group C of vectors called centroids. In a wav2vec, these quantized vectors are considered as the target of a transformer. As mentioned earlier, wav2vec uses a self-supervised strategy similar to BERT [4] for learning. This strategy involves randomly masking part of the feature encoder's output before sending it to the transformer, but the learning objective is formulated in a constrastive way and requires the identification of the correct representation, not of the encoded representation, but of the quantized latent audio representation q_t in a set of $K+1$ quantized candidate representations $\tilde{q} \in Q_t$ which include q_t and K distractors for each masked time step. The lost constrastive function can be expressed as follows: $-log\frac{exp(sim(c_t,q_t))}{\sum_{\tilde{q}\sim Q_t} exp(sim(c_t,\tilde{q}))}$ where c_t is the transformer output, and $sim(a,b)$ represents the cosine similarity. This loss is augmented by a codebook diversity penalty to encourage the model to use all codebook entries.

. To build a multilingual version of wav2vec 2.0, XLSR uses a shared quantization module on feature encoder representations, which means that feature encoder representations from different languages can be associated with the same quantized speech units. The multilingual quantized speech units produced by the quantization module are then used as targets for a transformer. This process forces the model to learn how to share discrete tokens between languages, creating a link between them that leads to a universalization of the acoustic representations

obtained by the model.

Decoder. The CTC algorithm was developped by Grave and al.[2] for labeling sequence data task. As we previously said, it is alignment-free ie in our case it doesn't require an alignment between the input vector segments produce by XLSR-53 and the output sentence segments. However, to get the probability of an output given an input, CTC works by summing over the probability of all possible alignments between the two. To define these possibles alignments, Grave et *al.* [2] introduce the ϵ symbol as a blank character in the output alphabet. This introduction solves two problems: (1) it is not logical to force each input step to align with an output in a speech recognition task; this symbol therefore marks a silence and (2) it marks the presence of several characters in a row, as it is difficult during recognition to know whether multiple identical letters in a row are a transcription of the same fragments or represent separate fragments, as is shown in figure 3 (a), putting an ϵ between them allows this difference to be made. As shown in figure 3 (a), a CTC alignment has the same length as the input, and we get the final output after merging the repeating characters and deleting the ϵ symbol. A CTC alignment is considered valid (figure 3 shows examples of valid and invalid CTC alignments for the "fas" output) for a given output if we can obtain the output from this alignment after the above-mentioned operations. CTC merges repeats characters between ϵ, so if an output has two of the same character in a row, then a valid alignment must have an ϵ between them. Based on previous description of alignment in CTC, during the training phase the objective is to maximize $P(Y|X) = \sum_{a \in A} \prod_{t=1}^{T} p(a_t|X)$ where a is a possible aligment and $p(a_t|X)$ is probability to have symbol a_t in time t in Y knowing X. $p(a_t|X)$ is given by the softmax at each time step. During inference phase CTC pick up $\hat{a} = \underset{Y}{argmax}(P(Y|a))$ as a final alignment and give an output after merging and remove operation.

. As mentioned earlier in our model, XLSR-53 is frozen during the train process ie only the weights of decoder are modified during the process. Once the model is trained, if we would like to use it to find a likely transcription for a given new raw speech data (waveform), we proceed as follow: encoded it by XLSR-53 in a vector X, then CTC decoder tent to provide $\hat{Y} = \underset{Y}{argmax}(P(Y|X))$ where $P(Y|X)$ is the probability to have a sentence Y with X as input. Then greedy search is used as an inference process to pick up $\hat{Y}$, meaning we take the letter with the highest probability at each time step, until you receive the special token symbolizing the end.

4.2 Experiment configuration

We have chosen three main axes of experiment, corresponding to different configurations of the feature extraction model and data pre-processing.

Tokenization. If a token for speech recognition is the character, Rolando Coto-Solono's work[9] on Bribri (a Latin American language), has shown that it could be beneficial in a tonale low ressources language context to make *tones explicit* in the transcriptions of texts to be recognized. In fact, he proposes to introduce tones as explicit characters to be recognized. To verify this aspect, in ours experiments we introduced two tokenization principles presented in Table 3: TonSep where tones are explicit characters to be recognized by the model, and ALL+tones where toned and character are considered as one symbol to reconized. However due to the extreme rarity of toned consonants in our dataset we have separed tones to them.

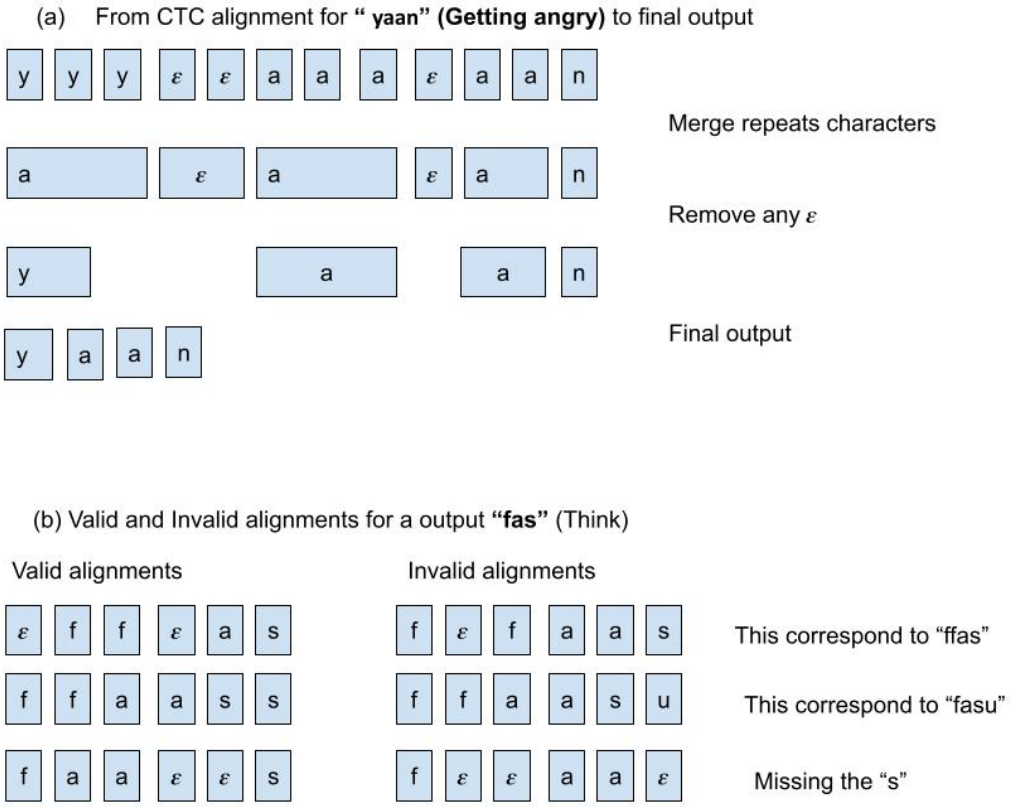

Figure 3: (a) Steps taken by CTC to obtain the final transcription of the word "yaan" from one of its valid alignments. Firstly, we merge the repeating characters that are not interspersed with ϵ and secondly, we delete ϵ. (b) Examples of valid and invalid raw output for the word "fas". An alignment is valid when we can obtain a correct final transcription after the operation described in (a)

.

Table 3: Type of tokenisation

Type de Tokenisation	Example	Tokenisation
ToneSep	ma wóg miǹtàg	m|a| w| ´|o|g |m|i| ̀|n|t|‘ |a|g|
All+tones	ma wóg miǹtàg	m|a|w|ó|g|m|i|‘|n|t|à|g|

Feature extractor . We have very little labelized data, so the XLSR-53 multilingual feature extraction model is frozen, which means that it provides vectors from the weights derived from its pre-training. We propose to experiment with various XLRS-53 pre-trained models. These models are presented in the Table 4. The model named *XLSR-fb*[2] is the standar model [7], LeBench[3] is the Wav2vec 2.0 LeBenchmark [11] trained on data from the French language exclusively; the remaining models (XLSR-kw and XLSR-sw) being produced from XLSR-fb by fine turning on a specified language, in fact these models was built using standard model weights as initial weights, then pre-training was continued using unlabeled data of a specific language (kinyarwanda,swahili). Following this method, *XLSR-kw*[4] is a specialized XLSR-53

[2]https://huggingface.co/facebook/wav2vec2-large-xlsr-53
[3]https://huggingface.co/LeBenchmark/wav2vec2-FR-7K-large
[4]https://huggingface.co/lucio/wav2vec2-large-xlsr-kinyarwanda

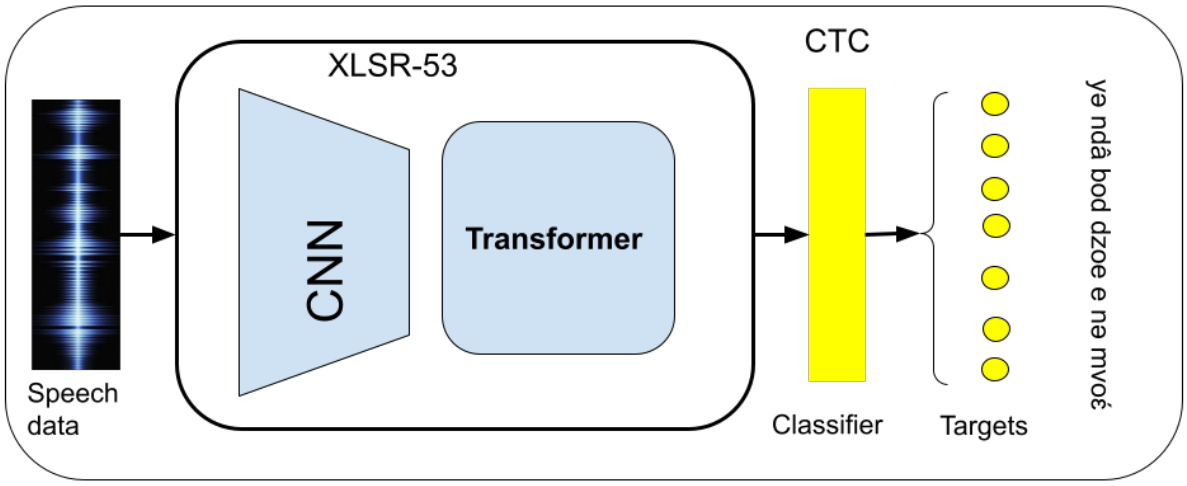

Figure 4: ASR with self-supervised XLSR-53 Model.
Speech data is passed in wavform to XLSR-53, which provides a vector representation of it. This representation is used by the CTC to predict a transcription.

model for the Kinyarwanda language and *XLSR-sw*[5] is a specialized XLSR-53 for the Swahili language, both of which are African bantue languages.

Table 4: Features extractions models

Model	Denomination	Source
XLSR-fb	Facebook XLSR-53	Hugging Face
LeBench	Wav2vec2 LeBench-mark	Hugging Face
XLSR-kw	XLSR kinyarwanda language	Hugging Face
XLSR-sw	XLSR for swahili language	Hugging Face

Language model. Previous ASR models required both a language model and a pronunciation dictionary to transform classified fragment sequences of audio recordings into a coherent transcript. Recent end-to-end models have made this possible, but [6] has shown that the use of a language model in conjunction with wav2vec 2.0 significantly improves ASR performance, especially in low ressources contexts. As part of our experiments, we tested the ASR model with the contribution of a bigram language model.

V EXPERIMENTS

The main objective of this work is to evaluate the performance of XLSR-53 for speech recognition of the Ewondo language. To achieve this goal, we collected and pre-processed speech data, then implemented the architecture described in Section 3. The choice of tools used to carry out

[5]https://huggingface.co/Akashpb13/Swahili_xlsr

these tasks was dictated by the literature. This section presents the details of these activities as well as the evaluation results.

5.1 Implementation details

Dataset and preprocessing. The Ewondo language has no public dataset for the ASR task, so we built a corpus from 103 sentences read by 5 speakers, including 4 men and one woman. We randomly selected 11 sentences for testing (2min30s) and the remaining 92 sentences for training (21min51s).The data was recorded at the computer science laboratory of Yaounde I, with a magnetophone, we use audacity[6] sofware for speech enhancement and artificially augmented these data with speechbrain[10] toolkit which using the method described in [**spec**] whose idea is to introduce noise into data using simple transformation on a signal like *speed perturbation, frequency dropout, time dropout* to obtain new data.

Architecture. We used the extraction models from the hugging face repository [7] [8] as well as the recipes proposed on the same platform for the development of the ASR model [8]. The model hyperparameters are the same as [12]. We have used the KenLM[3] framework to build the bigram language model using transcript texts only; this model simple store the probabilities of word pairs appearing in the transcripts.

5.2 Results and Discussion

Tables 6 and 5 show performances of the ASR model according to the different extraction models, but also according to the use of the language model (LM/no.LM) during decoding, and the use of artificial data (sA/no.SA). In these two tables, we can see that the performance associated with *Lebench* is by far the worst of all configurations. This discrepancy can be explained by two facts: Lebench is a monolingual extractor trained only on French, a language linguistically distant from Ewondo. We can also see from these tables that the use of the language model systematically increases the performance of the ASR model, which is consistant with the results presented in [6]. We also note the counter-intuitive results of artificial data augmentation, which degrades performance. One explanation for this contraction is to be found in the quantity of data from which the increase is made. Indeed, being of very small quantity, the artificial data acts as noise for the model.

Table 5: ASR model performance (%) with different feature extractors. Type of Tokenization = ALLfeat

	WER				**CER**			
	LM		no.LM		LM		no.LM	
	sA	no.sA	sA	no.sA	sA	no.sA	sA	no.sA
XLSR-fb	74.6	75.7	77.3	80.5	32.2	27.9	34.8	31.1
XLSR-rw	79.5	70.8	77.8	74.6	35.3	**28.6**	36.1	31
XLSR-sw	80	77.8	83.8	77.8	35.4	34.8	35.2	36.8
Lebench	97.3	97.3	100	100	93.9	97.3	100	100

Table 7 shows the average performance of the various ASR models in relation to the type of tokenization chosen. We can see that ToneSep is on average higher than ALL+tones, which means that it's better to recognize tones separately from characters in the low ressources case,

[6]https://www.audacityteam.org
[7]https://huggingface.co
[8]https://huggingface.co/blog/wav2vec2-with-ngram

Table 6: ASR model performance (%) with different feature extractors. Type of Tokenization = ToneSep

	WER				**CER**			
	LM		no.LM		LM		no.LM	
	sA	no.sA	sA	no.sA	sA	no.sA	sA	no.sA
xlsr-fb	77.3	73	89.4	82	32.7	27.6	36.2	29.8
xlsr-rw	78.4	77.3	88.8	87.1	33.6	29.8	36.5	31.2
xlsr-sw	79.5	75.1	87.1	75.1	33.2	30.2	36.2	32.3
Lebench	97.3	97.3	100	100	93.9	97.3	100	100

a result in line with the recommendations of [9]. On average, the XLSR-FB standars perform best (70.8% on WER and 28% on CER), outperforming the specialized models, which can be explained by the richness of their representation, However, overall performance remains low compared with the literature, which can be attributed to the extremely small amount of data available for training but also the distance existing between the target language and the languages underlying the pre-training of the acoustic model.

Table 7: Average results for each tokenization methods

	WER				**CER**			
	LM		no.LM		LM		no.LM	
	sA	no.sA	sA	no.sA	sA	no.sA	sA	no.sA
ToneSep	82.8	80.4	84.7	83.2	49.2	47.1	51.5	49.7
ALLfeat	83.1	80.6	91.3	86	48.3	46.2	52.2	48.3

5.3 Conclusion

The aim of this paper was to apply the multilingual acoustic model wav2vec XLSR-53 to the Ewondo language for the transcription task. Preliminary results show overall poor performance compared to the litterature in other languages (the best score being 70.8% on the WER and 28% on the CER). This result can be explain by the distance existing between the target language and the languages underlying the pre-training of the acoustic model. Although some similar work has already been carried out on African languages, our work reveals some singularities: firstly, the language of application, which to our knowledge is the first to be the subject of such a study; and secondly, the extremely small size of the dataset, which calls for greater finesse in pre-processing. In fact, in the literature working on low-resource data, datasets extend over at least several hours. This extremely low resource has enabled us to see the generalization limits of XLSR-53. Despite of the low performance, these experiments have enabled us to sketch out, apart from the need for additional data collection, some paths to follow in oder to improve the transcription model. The first is to pre-train a multilingual wav2vec XLSR-53 model on Ewondo recordings, in order to familiarize the model with the language; the second is to pay particular attention to the explicitness of tones in transcription, which has proved beneficial to the model; the third is to build a more robust language model from a richer corpus of text. To further evaluate wav2vec in the Ewondo transcription task, a comparison with others features extractors models is a particularly interesting prospect.

REFERENCES

Publications

[1] Z. Bao. "Moira Yip (2002). Tone. (Cambridge Textbooks in Linguistics.) Cambridge: Cambridge University Press. Pp. xxxiv+341." In: *Phonology* 20 (Aug. 2003), pages 275–279.

[2] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber. "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural 'networks". In: volume 2006. Jan. 2006, pages 369–376.

[3] K. Heafield. "KenLM: Faster and smaller language model queries". In: (July 2011).

[4] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. 2019. arXiv: 1810.04805 [cs.CL].

[5] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le. "SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition". In: *Interspeech*. 2019.

[6] A. Baevski, H. Zhou, A. Mohamed, and M. Auli. *wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations*. 2020. arXiv: 2006.11477 [cs.CL].

[7] A. Conneau, A. Baevski, R. Collobert, A. Mohamed, and M. Auli. *Unsupervised Cross-lingual Representation Learning for Speech Recognition*. 2020. arXiv: 2006.13979 [cs.CL].

[8] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, and A. M. Rush. *HuggingFace's Transformers: State-of-the-art Natural Language Processing*. 2020. arXiv: 1910.03771 [cs.CL].

[9] R. Coto. "Explicit Tone Transcription Improves ASR Performance in Extremely Low-Resource Languages: A Case Study in Bribri". In: Jan. 2021, pages 173–184.

[10] M. Ravanelli, T. Parcollet, P. Plantinga, A. Rouhe, S. Cornell, L. Lugosch, C. Subakan, N. Dawalatabad, A. Heba, J. Zhong, J.-C. Chou, S.-L. Yeh, S.-W. Fu, C.-F. Liao, E. Rastorgueva, F. Grondin, W. Aris, H. Na, Y. Gao, R. D. Mori, and Y. Bengio. *SpeechBrain: A General-Purpose Speech Toolkit*. 2021. arXiv: 2106.04624 [eess.AS].

[11] S. Evain, H. Nguyen, H. Le, M. Z. Boito, S. Mdhaffar, S. Alisamir, Z. Tong, N. Tomashenko, M. Dinarelli, T. Parcollet, A. Allauzen, Y. Estève, B. Lecouteux, F. Portet, S. Rossato, F. Ringeval, D. Schwab, and L. Besacier. "Modèles neuronaux pré-appris par auto-supervision sur des enregistrements de parole en français". In: *JEP 2022*. île de Noirmoutier, France, June 2022.

[12] C. Macaire, D. Schwab, B. Lecouteux, and E. Schang. "Automatic Speech Recognition and Query By Example for Creole Languages Documentation". In: *Findings of the Association for Computational Linguistics: ACL 2022*. Edited by S. Muresan, P. Nakov, and A. Villavicencio. Dublin, Ireland: Association for Computational Linguistics, May 2022, pages 2512–2520.

[13] A. Mohamed, H.-y. Lee, L. Borgholt, J. D. Havtorn, J. Edin, C. Igel, K. Kirchhoff, S.-W. Li, K. Livescu, L. Maaloe, T. N. Sainath, and S. Watanabe. "Self-Supervised Speech Representation Learning: A Review". In: *IEEE Journal of Selected Topics in Signal Processing* 16.6 (Oct. 2022), pages 1179–1210. ISSN: 1941-0484.

A ACKNOWLEDGEMENTS

This work has been funded by the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie grant agreement No 101007666, the Agency is not responsible for these results or use that may be made of the information.

B BIOGRAPHY

Yannick Yomie Nzeuhang was born in Douala, Cameroon in 1996. He received the B.S. degree in computer science from University of Douala, Cameroun, in 2017 and the M.S. degree in computer science from University of Yaounde I, Cameroon in 2020. He is currently pursuing the Ph.D. degree in Artificial intelligence in the same University. His research interests include machine learning, natural language processing and speech processing of low ressourced languages.

Paulin Melatagia Yonta was born in Garoua, Cameroon in 1980. He received a Ph.D. degree in computer science from the University of Yaounde I, Cameroon, in 2011. From 2008, he is a Researcher and Lecturer in the department of computer science of the same university. His research interests include machine learning, natural language processing and speech processing of low ressourced languages.He is an Invited Editor of the journal ARIMA. Dr Melatagia Yonta has been a member of the Council of the Research Unity of UMMISCO since 2021 and from 2022 the member of the executive committee of ASDS (The African Society in Digital Sciences) and the scientific secretary of the conference CRI (Conference on Research in Computer Science), from 2015.

Benjamin Lecouteux obtained a Master in 2005 in the field of Natural Language Processing . He joined the LIA in 2005 as Ph.D. student, in the Speech Processing group. Since 2011 he is Associate Professor at UGA (University Grenoble Alpes) and researcher at GETALP (Study Group for Machine Translation and Automated Processing of Languages and Speech)/LIG in Grenoble (France). Pr Benjamin Lecouteux received his HDR in 2021 and has been a professor since 2022. His current research interest are Language modeling for Automatic Speech Recognition, Machine Translation Systems, Neural Networks applied to NLP and more specifically, self-supervised models.