ARIMA

EPIsciences

Application of the multilingual acoustic representation model XLSR 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

DOI : 10.46298/arima.13621
Submitted on 20, May, 2024 - Published on 9, September, 2024
Volume : **Volume 42 - Special issue CRI 2023 - 2024** - Year : **2024**
Special Issue : **Volume 42 - Special issue CRI 2023 - 2024**
Editors : *Paulin*-1 *Melatagia*-1, *René*-2 *Ndoundam*-2, *Kamel*-2 *Barkaoui*-2, *Blaise Omer*-2 *Yenke*-2

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 and wav2vec 2.0 XLSR-128, for the transcription task of the Ewondo language (spoken in Cameroon). The experiments were conducted on 11 minutes of speech recorded from 103 read sentences. Despite a strong generalization capacity of multilingual acoustic model, preliminary results show that the distance between XLSR embedded languages (English, French, Spanish, German, Mandarin, . . .) and Ewondo strongly impacts the performance of the transcription model. The highest performances obtained are around 69% on the WER and 28.1% 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 ; 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** [8] and also **XLSR-128** [10], which deliver impressive results for automatic speech recognition (ASR) tasks, even on small datasets. By these fact XLSR 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 lack of 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*[7] in various configurations, we have evaluated the performance on word error rate (WER) and character error rate (CER). Our contribution in this paper is twofold: 1) The construction of a basic speech recognition model for 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 organized 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] [4] 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

Pontuel tone		Modular tone	
Denomination	Notation	Denomination	Notation
Low [Tb]	[v̀]		
High [HT]	[v́]	High-Low [HLT]	[v̂]
Medium [MT]	[v̄]		
M-Low [MLT]	[v]		
Supra-High [SHT]	[v̋]	Low-High [LHT]	[v̌]
Infra-Low [SIL]	[v]		

Table 1: Tones in Ewondo language

Words	Translation	Words	Translation
minkud	*bag*	*minkúd*	*cloud*
zám	*raffia*	*zàm*	*good taste*
bám	*to scold*	*bam*	*to worry*
bóg	*to pile up*	*bog*	*to extract*
tag	*to rejoice*	*tág*	*to classify*

Table 2: Words that differ only in tone

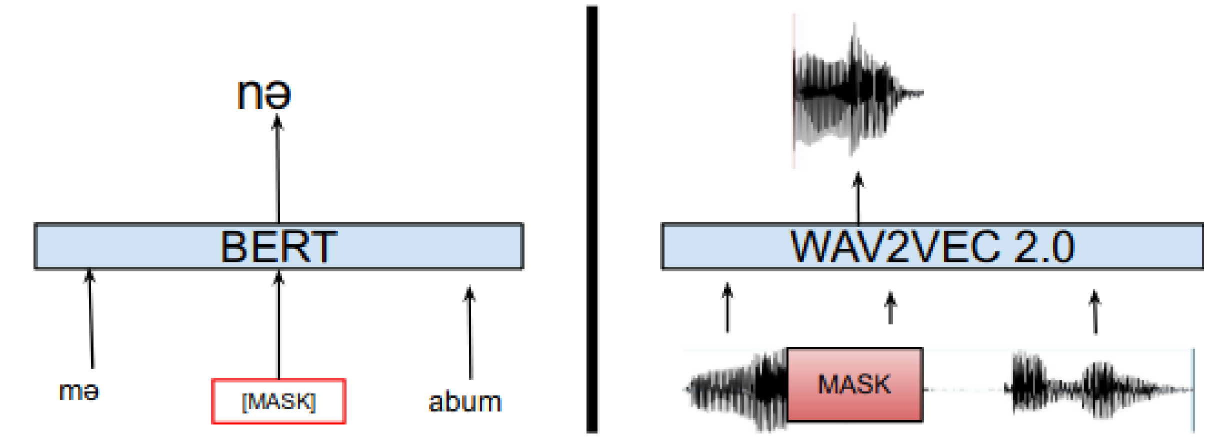

Figure 1: **Left**. Example of a BERT pre-training task with the sentence "m n abum" (I'm pregnant), the word "n" is hidden and the model must predict it. **Rigth**. The same concept is applied to the audio signal, where certain portions are masked and wav2vec 2.0 must predict them.

2.2 ASR with Self-supervised models

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 use large 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. 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 [16]), but we have chosen to exploit the XLSR-53 a crosslingual version of wav2vec 2.0 [7] for its promising results on languages with small amounts of data. This model uses a pre-training task similar to BERT[6], illustrated in Fig 1. 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 divide our model in two parts; the cross-lingual speech representations (XLSR) [8, 10] as a feature extractor and connectionist temporal classifier(CTC) [2] as a classifier. This architecture is mainly inspired by that of [15], used to exploit self-supervised models for two Creole languages, but also by [8, 10, 11] which used the same architecture to demonstrate the effectiveness of XLSR. This section present the overall design of the model and different configurations used during experiments.

3.1 The 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 3 we use as feature extractor the cross-lingual speech representations (XLSR)[8] version which is a multilingual representation model pre-trained on many 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 will not be modified during training. Our architecture is a same as [15], an encoder-decoder architecture where XLSR 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) 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 each part of our model.

Encoder. This XLSR 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. As mentioned earlier, wav2vec uses a self-supervised strategy similar to BERT [6] 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.

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 i.e in our case it doesn't require an alignment between the input vector segments produce by XLSR 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. CTC merges repeats characters between ϵ (Figure 2), so if an output has two of the same character in a row,

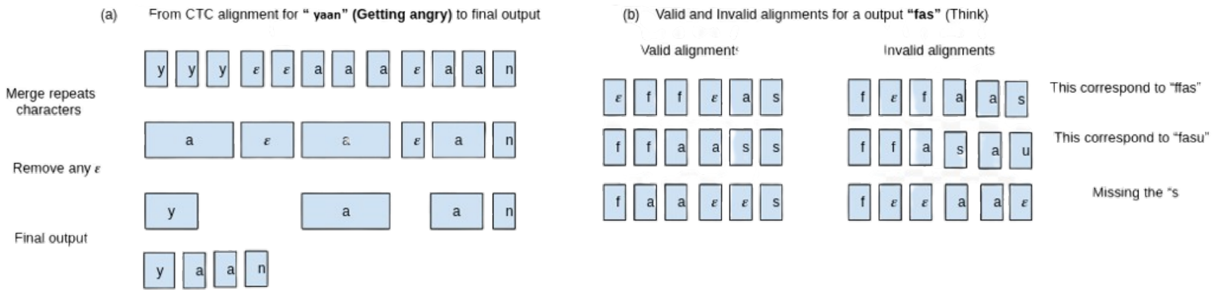

Figure 2: (a) Steps taken by CTC to obtain the final transcription of the word "yaan" from one of its valid alignments. Firstly, we merge the repeating characters that are not interspersed with ϵ and secondly, we delete ϵ. (b) Examples of valid and invalid raw output for the word "fas". An alignment is valid when we can obtain a correct final transcription after the operation described in (a)

.

then a valid alignment must have an ϵ between them (Figure 2). Based on previous description of alignment in CTC, during the training phase the objective is to maximize

$$P(Y|X) = \sum_{a \in A} \prod_{t=1}^{T} p(a_t|X) \tag{1}$$

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 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 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.

3.2 Experiment setup

We have chosen three main axes experiments, corresponding to different configurations of the features extractor model and data pre-processing.

Tokenization. If a token for speech recognition is the character, Rolando Coto-Solono's work[12] 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 symbols to be recognized by the model, and ALL+tones where a tone was associated to a character and represented as one symbol to reconized. We also used *byte pair encoding* (BPE)[5] , a popular machine translation technique for tokenization that subdivides words into subunits to keep vocabulary as small as possible.

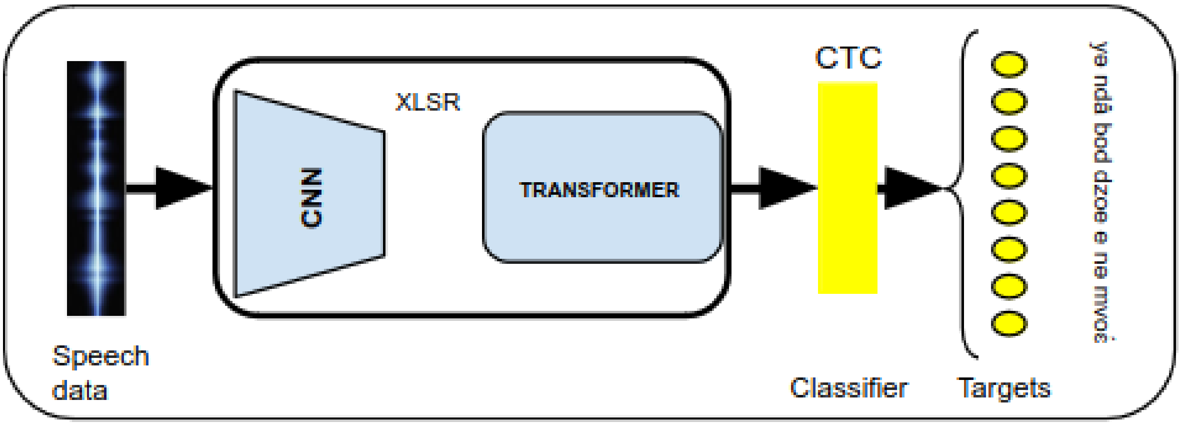

Figure 3: ASR with self-supervised XLSR Model. Speech data is passed in wavform to XLSR, which provides a vector representation of it. This representation is used by the CTC to predict a transcription.

<table>
<tr><th>Type of Tokenisation</th><th>Example</th><th>Tokenization</th></tr>
<tr><td>ToneSep</td><td>ma wóg mìǹtàg</td><td>m|a| w|´|o|g |m|i| ̀n|t|` |a|g|</td></tr>
<tr><td>All+tones</td><td>ma wóg mìǹtàg</td><td>m|a|w|ó|g|m|i|‘|n|t|à|g|</td></tr>
</table>

Table 3: Type of tokenization

Features extractor . We have very little labelized data, so the XLSR multilingual features 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 pre-trained models. These models are presented in the Table 4. Models named *XLSR-300m*[2], *XLSR-1b*[3] *XLSR-2b*[4] are respectivily the standard XLSR-128 model with 300 million, 1 bilion and 2 bilion of parameters; *XLSR-53*[5] is the standar model [8], LeBench[6] is the Wav2vec 2.0 LeBenchmark [14] trained on data from the French language exclusively; the remaining models (XLSR-kw and XLSR-sw) being produced from XLSR-53 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,and swahili). Following this method, *XLSR-kw*[7] is a specialized XLSR-53 model for the Kinyarwanda language and *XLSR-sw*[8] is a specialized XLSR-53 for the Swahili language, both of which are African bantue languages.

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 [7] 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 (this bigram is use to produce tables results 5,6,8) constructed from the transcriptions of the recordings in our dataset and also a 5-gram (used to produce the result of table 9) constructed on new testament of the Bible .

[2]https://huggingface.co/wav2vec2-xls-r-300m
[3]https://huggingface.co/wav2vec2-xls-r-1b
[4]https://huggingface.co/wav2vec2-xls-r-2b
[5]https://huggingface.co/facebook/wav2vec2-large-xlsr-53
[6]https://huggingface.co/LeBenchmark/wav2vec2-FR-7K-large
[7]https://huggingface.co/lucio/wav2vec2-large-xlsr-kinyarwanda
[8]https://huggingface.co/Akashpb13/Swahili_xlsr

Model	Denomination	Source
XLSR-300	Facebook XLSR-300m	Hugging Face
XLSR-1B	Facebook XLSR-300m	Hugging Face
XLSR-2B	Facebook XLSR-300m	Hugging Face
XLSR-53	Facebook XLSR-53	Hugging Face
LeBench	Wav2vec2 LeBenchmark	Hugging Face
XLSR-kw	XLSR kinyarwanda language	Hugging Face
XLSR-sw	XLSR for swahili language	Hugging Face

Table 4: Features extractions models

IV EXPERIMENTS

The main objective of this work is to evaluate the performance of XLSR 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 literature has helped us choose the right tools to carry out these tasks. This section presents the details of these activities as well as the evaluation results.

4.1 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.To ensure that there was no utterance for a sentence in both the training and test sets, we randomly selected 10% of the sentences for the test (1min3s) and the remaining 90% of the sentences for the training (9min51s). The data was recorded at the Computer Science Laboratory of Yaounde I, with a dictaphone, we use audacity[9] sofware for speech enhancement.

Architecture. We used the extraction models from the hugging face repository [10] [9] as well as the recipes proposed on the same platform for the development of the ASR model [11]. The model hyperparameters are the same as [15]. We have used the KenLM[3] framework to build the bigram language model using transcript texts only and 5-gram using a New Testament Bible; this model simply stores the probabilities of word tuples appearing in the text.

4.2 Results and Discussions

Tables 5,6, and 7 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. In these 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 [7]. This observation is confirmed by the table 9, which shows an increase in the average performance of the 4 models tested with an LM (5-gram) built on a larger corpus. However, given the difference in the amount of text used to produce the bigram and the 5-gram, we might have expected a more significant difference in the performance of the models that exploit them. One hypothesis that could explain these results is the context of the text corpus chosen to build the 5-gram (Bible),

[9] https://www.audacityteam.org
[10] https://huggingface.co
[11] https://huggingface.co/blog/wav2vec2-with-ngram

	WER		CER	
	LM	no.LM	no.LM	LM
XLSR-300m	789	85	41	36
XLSR-1b	78	84.1	46.2	41.8
XLSR-2b	78.9	91.7	46.1	41.3
XLSR-53	80	85.2	37.7	33.6
XLSR-rw	80	81.1	36.5	36.4
XLSR-sw	81.6	86.4	40.2	37.3
Lebench	97.3	1	1	93

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

	WER		CER	
	LM	no.LM	no.LM	LM
XLSR-300m	69	72.4	31.5	28.1
XLSR-1b	77.8	78.9	36.2	36
XLSR-2b	79.4	82.1	36.9	33.5
XLSR-53	80	85.2	37.7	33.6
XLSR-rw	75.6	82.1	31.9	30
XLSR-sw	74	75.1	30.2	28.3
Lebench	97.3	1	1	93

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

	WER		CER	
	LM	no.LM	no.LM	LM
XLSR-300m	96.7	84.1	35.4	84.4
XLSR-1b	97.8	100	69.6	69.4
XLSR-2b	94.5	100	72.3	86.3
XLSR-53	97.8	100.0	63.9	70
XLSR-rw	97.2	100	61.1	63.0
XLSR-sw	97.2	100	74.2	67.1
Lebench	97.3	1	1	93

Table 7: ASR model performance (%) with different feature extractors. Type of Tokenization = BPE

	WER		CER	
	LM	no.LM	no.LM	LM
ToneSep	79	82	43	40.3
ALL+tones	82.2	87	53.7	76.1
BPE	96	97	68	76

Table 8: Average results for each tokenization methods

which is far removed from the transcription text and therefore not sufficiently helpful during decoding. We also note the counter-intuitive results of the large XLSR-128 (XLSR-1b and XLSR-2b), which performs less well than XLSR-300m. One explanation might be that the representations produced by *XLSR-1b, XLSR-2b* are proportional to the width of the model, and that the larger the representation, the more data is needed to finetune the decoder.

Table 8 shows the average performance of the various ASR models in relation to the type of tokenization chosen. The BPE acts as the worst of all, suggesting that it is not suited to this task; on the other hand, 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, a result in line with the recommendations of [12]. On average, the XLSR-300m performs best (69% for the WER and 28.1% for the CER), outperforming the specialized models, which can be explained by the richness of their representation due to the great diversity of languages used for pre-training. However it is following by the XLSR-sw (74% on WER and 28.3% on CER) and XLSR-rw (75.6% on WER and 30% on CER) respectively, this can be explained by the proximity of these two languages to Ewondo. 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.

4.3 Conclusion

The aim of this paper was to apply the multilingual acoustic model wav2vec XLSR 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 69% on the WER and 28.1%

	WER			CER		
	ALL+tones	ToneSep	BPE	ALL+tones	ToneSep	BPE
XLSR-300m	83.2	70.8	91.8	37.6	28.1	90.4
XLSR-53	80	73.51	90.8	33.6	290	85.9
XLSR-rw	82.1	77.2	100	34.4	30.5	60.9
XLSR-sw	77.2	72.4	100	66.9	30.9	35

Table 9: ASR model performance (%) with different feature extractors. LM=5-gram

on the CER). These results can be explain by very small size of the dataset and 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. 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 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 consists in building a more robust linguistic model from a richer corpus of texts that is also close to the transcription corpus. To further evaluate wav2vec in the Ewondo transcription task, a comparison with others features extractors models is a particularly interesting prospect.

V 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.

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.-M. ESSONO. *Langue et culture ewondo: par la grammaire, les textes et l'exercice : (suivi d'un lexique français-ewondo)*. Mar. 2012.

[5] R. Sennrich, B. Haddow, and A. Birch. *Neural Machine Translation of Rare Words with Subword Units*. 2016. arXiv: 1508.07909 [cs.CL].

[6] 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].

[7] 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].

[8] 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].

[9] 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].

[10] A. Babu, C. Wang, A. Tjandra, K. Lakhotia, Q. Xu, N. Goyal, K. Singh, P. von Platen, Y. Saraf, J. Pino, A. Baevski, A. Conneau, and M. Auli. *XLS-R: Self-supervised Cross-lingual Speech Representation Learning at Scale*. 2021. arXiv: 2111.09296 [cs.CL].

[11] A. Conneau, A. Baevski, R. Collobert, A. Mohamed, and M. Auli. "Unsupervised Cross-Lingual Representation Learning for Speech Recognition". In: *Proc. Interspeech 2021*. 2021, pages 2426–2430.

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

[13] 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].

[14] 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.

[15] 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.

[16] 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.