INTERSPEECH 2023
20-24 August 2023, Dublin, Ireland

Exploring the Impact of Pretrained Models and Web-Scraped Data for the 2022 NIST Language Recognition Evaluation

Tanel Alumäe[1], *Kunnar Kukk*[1], *Viet-Bac Le*[2], *Claude Barras*[2], *Abdel Messaoudi*[2], *Waad Ben Kheder*[2]

[1]Tallinn University of Technology, Estonia
[2]Vocapia Research, France

{tanel.alumae,kukkar.kukk}@taltech.ee, {levb,barras,abdel,benkheder}@vocapia.com

Abstract

This paper describes Vocapia-TalTech team systems developed for the 2022 NIST Language Recognition Evaluation (LRE22) which focused on spoken language identication of African languages, including low-resource languages. In both fixed and open conditions, our primary systems were fused from multiple individual systems using logistic regression. In the fixed condition, we largely relied on wav2vec2.0 conformer models pretrained on the provided training data. In the open condition, we used external pretrained wav2vec2.0 models, phonotactic models and features derived from a multilingual speech recognition system, and also augmented the provided target language development data with additional data scraped from the web. On the LRE22 evaluation data, our final fixed and open condition systems obtained excellent results, with primary metric C_{act} values of 0.111 and 0.067, respectively. A post-evaluation study shows that both pretrained models as well as additional data are important for accurate models.

Index Terms: spoken language recognition, NIST LRE22, wav2vec2.0

1. Introduction

The identification of spoken language (LID) involves recognizing the language spoken in an audio segment. In recent years, utilizing pre-trained models based on self-supervised or speech recognition objective functions has emerged as a state-of-the-art approach for LID, as evidenced by prior research [1, 2].

The 2022 NIST Language Recognition Evaluation (LRE22) [3] focused on LID for 14 African languages and dialects. The task can be considered low-resource, as speech from only 30 speakers for each target language was available for system development. Thus, LRE22 provided a good opportunity to experiment with various pre-trained models. This paper presents systems developed for LRE22 by the Vocapia-TalTech team, along with a description of the additional training data we collected from the web and its impact on system performance.

2. NIST LRE22

LRE22 was the ninth evaluation in a series that began in 1996. The evaluation aims to advance LID technology, facilitate its development and measure its performance. LRE22 focuses on conversational telephone speech and broadcast narrow band speech data in the context of low-resource training data, with an emphasis on African languages. The task for LRE22 is closed-set language detection: given a segment of speech and a target language, automatically determine if the target language was spoken in the test segment. LRE22 has 14 target languages: Afrikaans, Tunisian Arabic, Algerian Arabic, Libyan Arabic, South African English, Indian-accented South African English, North African French, Ndebele, Oromo, Tigrinya, Tsonga, Venda, Xhosa and Zulu.

In the fixed condition, participants are allowed to use the following speech datasets for system training and development:

- NIST LRE 2017 Development Set, previous NIST LRE training data (LDC2022E16), NIST LRE 2017 Test Set (LDC2022E17), containing 1872 h of conversational telephone speech from 14 languages/dialects;
- NIST LRE 2022 Development Set (LDC2022E14), containing 26 h of telephone speech from 14 target languages;
- VoxLingua107 [4], containing 6628 h of Youtube speech from 107 languages.

Only the LRE22 development set contains speech data for all 14 target languages, with 300 segments per language, collected from 30 unique speakers per language. LRE17 datasets also contain mostly conversational telephone speech, but for a different set of languages and dialects. In the open training condition, participants can use any additional data including proprietary data.

The primary metric for LRE22 is an application-motivated cost function C_{act} that combines the probability of missed detection and false alarms, given log-likelihood scores assigned to each language by the system [5]. Since LRE22 considers closed-set language classification, we also provide simple classification error rates in addition to the official metric scores.

3. Fixed condition

3.1. Voice activity detection

We used Kaldi's [6] energy-based voice activity detection for processing both training as well as test data. Frame-based voice activity decisions were used as input to a process that creates longer speech chunks by finding contiguous sequences of voice frames and then merging them using a criterion that observes the proportion of voiceless frames in a chunk. Resulting long training data speech segments were split into 10-second segments, with an overlap of 2 seconds.

3.2. Frontends

3.2.1. *Resnet-style model*

The Resnet-style model is derived from the x-vector paradigm [7, 8], with several enhancements. During training, we apply on-the fly data augmentation, by randomly distorting the training data using reverberation and noise augmentation. We used the background noises in the Freesound portion of the MUSAN corpus [9] and simulated small, medium and large room impulse responses [10] for data augmentation.

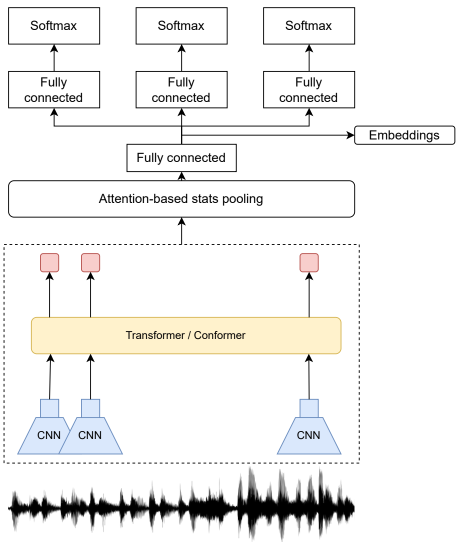

Figure 1: *Wav2vec2.0-based model with block-softmax output layers.*

For frame-level feature extraction, we use the Resnet34 [11, 12] architecture where the basic convolutional blocks with residual connections are replaced with squeeze-and-excitation modules [13, 14]. The statistics pooling layer that maps frame-level features to segment level features is replaced in our model with a multi-head attention pooling layer [15]: 512-dimensional frame level representations are first mapped to 128 outputs, using a 1×1 convolution and a ReLU nonlinearity; from this representation, each attention head (we used 5 heads) computes it's own softmax-based weight distribution over the input utterance; finally, weighted mean and standard deviation are computed over the frame level features for each head, resulting in $5 \times 512 \times 2$ segment-level representations.

The Resnet model was trained on the full VoxLingua107 training set, resampled to 8 kHz. 30-dimensional filterbank features were used as input.

3.2.2. *Wav2vec2.0-based frontends*

Since using external pretrained models is not allowed in the fixed track, we trained our own wav2vec2.0 [16] models on the provided training data. We used the "conformer-base" architecture to train two models: one on the LRE17 training and development data, an the other on the union of LRE17 and VoxLingua107. The VoxLingua107 data was band-filtered to simulate 8 kHz data.

The wav2vec2.0 models were then finetuned for LID as follows: the outputs from the wav2vec2.0 "backbone" were fed through an attentive pooling layer, a fully connected layer with ReLU and BatchNorm, and the final output layer, corresponding to the languages of the training set. During training, the learning rate corresponding to the backbone was set to 1% of the base learning rate. We experimented with two finetuning scenarios: (1) using VoxLingua107, (2) using both LRE17 and VoxLingua107, using block-softmax. In the block-softmax model, different output layers are created for different training datasets (see Figure 1). This is different from the usual approach where all languages from all training datasets are pooled together during training. This approach has been used in training multilingual bottleneck extractors [17] and speaker recognition models [18]. It is motivated by the fact that the language data in the two training datasets (VoxLingua107, LRE17) is quite different. By separating them into different softmax blocks, we encourage the model to learn dataset independent features in the shared layers (including the embedding layer) and place the dataset-specific discrimination capability into the final branched layers. Also, this eliminates the problem of how to model dialects/languages that have a different granularity in different training datasets – e.g., different English and Arabic dialect/accents are individual target languages in LRE17 but not in VoxLingua107.

The same on-the-fly data augmentation strategy as was used in training the Resnet-style model was also used in finetuning wav2vec2.0 based models.

3.2.3. *X-vector model, using wav2vec2.0 features*

The x-vector model [7] consists of a feed-forward DNN that maps sequences of variable-length speech features to fixed-dimensional embeddings. In language recognition, multilingual bottleneck features have been found to perform well as input features to an x-vector model [8]. In the closed condition, instead of using phonetic bottleneck features, we extracted representations from the last layer of the frozen wav2vec2.0 models described in the previous section. The dimension of the extracted wav2vec2.0 features is 768. The original TDNN x-vector architecture [7] was used. The x-vector dimension (for the backend) is 128.

3.3. Backend

Most of the individual systems used a backend that consists of LDA (dimensionality of 13), mean subtraction, length normalization, and logistic regression. It is trained on noise, music and reverberation augmented LRE22 development data, after voice-activity detection and chunking, as described in Section 3.1.

The backend for x-vector systems computes the log posterior probabilities for each target language directly with a LDA trained on mean and variance-normalized x-vectors.

4. Open condition

4.1. Additional datasets

In the open track, participants were allowed to use additional data for system training and development. We collected two web-scraped datasets containing target language data (Table 1).

The first dataset Extra$_A$ was collected in a rapid manner. For most target languages, we found radio stations that mostly broadcast in the specific language that also provide links to episodes of specific radio programs via RSS or YouTube. We tried to find specific radio programs under each station that have a high presence of conversational speech and a large variety of speakers. We didn't attempt to verify that the individual downloaded episodes contain the target language. For two languages/dialects (Algerian and Libyan Arabic), we simply reused data from the ADI17 corpus [19] which contains data originating from Youtube. The retrieved data was segmented into single-speaker speech segments using *pyannote.audio* [20, 21] and further subsegmented into consecutive 10-second segments. We finally reduced the dataset to 30k segments per language. This approximates to around 50 hours per language, with the exception of South African English and Ndebele, for which we had less data downloaded.

When collecting the second dataset Extra$_B$, we put more emphasis on collecting conversational telephone speech and ex-

Table 1: *Additional web-scraped datasets used in the open condition (before speech detection)*

	Rapidly collected dataset Extra$_A$		**Carefully collected dataset Extra$_B$**	
Language	Sources	Hours	Sources	Hours
Afrikaans	RSG (radio), Klipkouers (Podcast)	165	RSG (radio), SABC (TV)	61
Tunisian Arabic	Watania (TV)	280	Mosaique FM, Diwan FM	48
Algerian Arabic	ADI17 (Youtube)	51	Darba Jil FM, JowRadio Algerie	29
Libyan Arabic	ADI17 (Youtube)	80	Waad FM, Libya Mostakbal (radio)	20
South-African (SA) English	SaFM (radio)	21	Sa FM, Man Torg, Radio 702 (radio)	62
Indian-accented SA English	Lotus FM (radio)	29	IndiaToday (radio), NewsOnAir (radio)	57
North African French	Radio Algerienne FM (radio)	47	France Maghreb (radio), Radio M	61
Ndebele	Indaba zesiNdebele (radio program)	25	IKwekwezi FM , SABC (TV)	60
Oromo	United Oromia (Youtube)	67	Oromo News (TV), VOA Oromo (TV)	50
Tigrinya	Tigrai TV (TV station)	129	Radio Erena	50
Tsonga	Munghana Lonene FM (radio)	75	Munghana Lonene FM, SABC (TV)	48
Venda	Phalaphala FM (radio)	52	Phalaphala FM, SABC (TV)	48
Xhosa	Umhlobo Wenene FM (radio)	52	Umhlobo Wenene FM , SABC (TV)	59
Zulu	Ukhozi FM (radio)	234	Ukhozi FM, SABC, IARPA Babel (LDC2017S19)	60

cluding non-target language data. Additional web sources were identified for this collection. Collected data was automatically processed through SAD, segmentation and filtered to exclude other languages according to the following procedure:

1. Using a multi-domain SAD system to detect speech segments from each source corresponding to telephone speech (additional broadcast speech segments are added in case not enough telephone speech are found in the original data).
2. Filtering out segments belonging to foreign languages (mainly English and French segments for African languages and Arabic dialects). This was done using intermediate phonotactic and/or x-vector models. The accented English is the only exception to this filtering process where all segments detected as English from South African and Indian sources (using our internal LID system) are simply kept without any further analysis.
3. Building a model based on a subset of segments corresponding to each language and removing segments belonging to each class with a low confidence measure. A cross-validation procedure is used in this step to avoid any source of bias.
4. Manually checking the remaining segments and keeping a subset of 300 segments per language (or dialect) with speech duration between 3 and 30 seconds. It's important to mention that the filtering of Arabic segments should be done carefully to remove any segments that have too much MSA or code-switching (mainly French) and only keep good-quality segments to be used in the final LID system.

On the average, 30-40 hours per language or dialect were selected from telephone (narrowband) speech and 20 hours from broadcast (wideband) speech. For Zulu, we also included the corresponding IARPA Babel Language Pack.

4.2. Wav2vec2.0 based subsystems

In the open condition, our submission used two publicly available wav2vec2.0 models pretrained on around 500 000 hours of unlabelled data: XLS-R-1b and XLS-R-2b [22]. The models were finetuned for LID with block-softmax over three datasets (VoxLingua107, LRE17, and Extra$_A$) and then optionally further finetuned on the Extra$_B$ dataset.

The backend is similar as was used in the fixed condition, except that it is trained on the union of data-augmented and chunked LRE22 development data and the new dataset Extra$_B$.

4.3. X-vector model, using ASR bottleneck features

For open condition, at input of the x-vector models, we used the representations extracted from the pre-final layer of a multilingual ASR TDNN-F model. However, instead of the DNN bottleneck model, we used a conventional phonetic TDNN-F model [23] without any specific bottleneck layer. The ASR TDNN-F model was trained on 950 hours of conversational telephone speech data, including BABEL, Appen, and Linguistic Data Consortium (LDC) data, covering 17 languages. A common phoneset comprising 68 phones and 3 non-speech units was used.

A TDNN x-vector model with about 18M parameters was trained on the Extra$_B$ corpus using 192-dimensional bottleneck features, extracted from the pre-final TDNN layer of the ASR model. The x-vector dimension is set to 128.

4.4. Phonotactic models

Phonotactic systems rely on the assumption that the phonotactic characteristics, that is the way phonemes make up words and sentences, differ across languages [24, 25].

The first phonotactic system used in our experiments makes use of the Parallel Phone Recognizer followed by Language Modeling (PPRLM) approach [25]. Pre-trained phone decoders using TDNN-F acoustic models for three languages (English, French and Arabic) were used to decode all of the Extra$_B$ data. Phone n-gram statistics were then estimated from the resulting phone lattices and used to compute the expectation of the phone log-likelihood for each target language [26]. The PPRLM system merges some dialects and languages into single targets and thus its results are not directly comparable to other systems. It was primarily designed for fusion with acoustic models.

An alternative phonotactic model relies on a pretrained multilingual phone recognizer [27][1], trained on the phoneme transcripts of 21 Mozilla CommonVoice datasets and 19 BABEL corpora. It uses a XLSR-53 wav2vec2.0 model [28] as initialization. For training the phonotactic system, we used the multilingual phone recognizer to transcribe LRE22 development set and the web-scraped training dataset Extra$_B$. Based on the resulting transcriptions, we trained a multinomial Naive Bayes model that uses all phoneme bigrams, trigrams and 4-grams that occur in training data as features.

[1]Available at `https://huggingface.co/facebook/wav2vec2-xlsr-53-phon-cv-babel-ft`

Table 2: *Results of individual systems and their fusion on LRE22 evaluation set.*

	ID	Frontend	Backend	Dev C_{act}	Eval C_{act}	Eval Err%
Fixed condition	F_1	Resnet, trained on VoxLingua107	LDA+LR	0.220	0.203	19.7
	F_2	Conformer w2v, pretrained on LRE17, finetuned on VoxLingua107	LDA+LR	0.153	0.135	15.0
	F_3	Conformer w2v, pretrained on LRE17+VoxLingua107, finetuned on VoxLingua107	LDA+LR	0.166	0.151	15.0
	F_4	Like above, but finetuned on LRE17+VoxLingua107	LDA+LR	0.168	0.139	13.9
	F_5	Conformer w2v, pretrained on LRE17+VoxLingua107 $\rightarrow$ TDNN x-vector	LDA post.	0.187	0.173	17.9
	F_P	**Fusion**		**0.136**	**0.111**	**11.5**
Open condition	O_1	XLS-R-1B, finetuned on VoxLingua107+LRE17+Extra$_A$ + more ft. on Extra$_B$	LDA+LR	0.082	0.067	6.3
	O_2	XLS-R-2B, finetuned on VoxLingua107+LRE17+Extra$_A$	LDA+LR	0.090	0.075	7.7
	O_3	Multilingual phonetic TDNN-F features + TDNN x-vector	LDA post.	0.149	0.134	12.5
	O_4	Lattice-based PPRLM phonotactic system		0.327	0.320	34.8
	O_5	Multilingual phone recognizer based phonotactic system		0.350	0.368	29.9
	O_S	Fusion		0.075	0.067	6.3
	O_P	**Fusion, with duration-based features**		**0.074**	**0.067**	**6.4**

Table 3: *Impact of pretrained models and extra data.*

XLS-R	Finetuning data (in addition to VoxLingua107 and LRE17)	Backend training data	Eval C$_{act}$
300M	-	LRE22	0.109
1B	-	LRE22	0.107
1B	-	LRE22, Extra$_A$	0.101
1B	-	LRE22, Extra$_B$	0.096
1B	Extra$_A$	LRE22, Extra$_A$	0.069
1B	Extra$_A$	LRE22, Extra$_B$	0.071
1B	Extra$_A$, more ft. with Extra$_B$	LRE22, Extra$_B$	0.067

5. Results and analysis

The results of our systems and their fusion are listed in Table 2. Results on the development set are computed using 5-fold cross-validation. Both primary metric C_{act} and error rate scores are provided for the evaluation set.

Individual system scores were fused and calibrated using a logistic regression model trained on development data. The calibration model learns a scale and bias for each model and uses binary reference values as target values.

In the open condition, we experimented with duration-sensitive fusion. Each model's scores are fed to the calibration model twice: (1) as is, and (2) scaled by the log duration of the trial audio. This accounts for the idea that different models may have duration-specific importance. Duration-sensitive fusion improved results on development data but not on evaluation data. Bold scores from the fusion models were used as our primary LRE22 submissions.

Figure 2 shows the C_{act} scores of top-ranked teams in both conditions, with team names anonymized (copied from [3]). LRE22 rules prevent us from pinpointing our results in the ranking, but readers can compare the results in the figure to the evaluation set C_{act} scores shown in Table 2.

The results indicate that wav2vec2 models fine-tuned for LID outperformed other approaches. Although the conformer-based wav2vec2.0 model pre-trained on the provided training data outperformed all other models under fixed conditions, replacing it with the pre-trained XLS-R-1b model and using more target language data for fine-tuning led to a significant improvement of about 50% from F_2 to O_1. An analysis of fine-tuning XLS-R wav2vec2.0 models using varying amounts of web-scraped extra data is presented in Table 3. The first line corresponds to the XLS-R-300M model fine-tuned for language

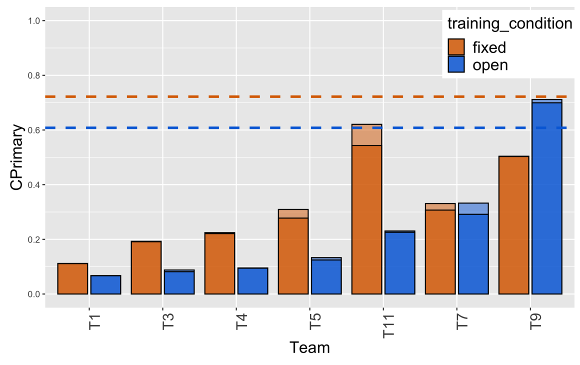

Figure 2: *LRE22 performance comparison of fixed and open training conditions, with anonymized team names [3]. Dashed lines correspond to the official baseline.*

recognition using VoxLingua107 and LRE17 data, with a backend classifier trained exclusively on LRE22 development data. Results show that using the larger 1B parameter model only marginally improved accuracy compared to the 300M model. Additionally, including extra web-scraped data for training the backend had a relatively small impact. However, utilizing the extra data to fine-tune a new frontend model resulted in an improvement of approximately 30%. The results also indicate that while the carefully curated web-scraped dataset Extra$_B$ generally yields better results than the rapidly collected Extra$_A$, the difference is relatively minor. This suggests that rapidly collecting somewhat weakly-labelled data is adequate for most practical purposes.

6. Conclusion

This paper outlined Vocapia-TalTech team systems developed for the NIST LRE22. Leveraging pretrained models based on wav2vec2.0 proved to be crucial for achieving very strong results in both fixed and open conditions. Specifically, in the fixed condition, we employed conformer-based wav2vec2.0 models pretrained on the provided speech data. In the open condition, we obtained a 30% performance improvement by utilizing publicly available XLS-R models. Moreover, we fine-tuned wav2vec2.0 models using target language data rapidly scraped from the web, which led to an additional 30% boost in results.

7. References

[1] A. Lyu, Z. Wang, and H. Zhu, "Ant multilingual recognition system for OLR 2021 Challenge," in *Interspeech*, 2022, pp. 3684–3688.

[2] T. Alumäe and K. Kukk, "Pretraining approaches for spoken language recognition: TalTech submission to the OLR 2021 Challenge," in *The Speaker and Language Recognition Workshop (Odyssey 2022)*, 2022, pp. 240–247.

[3] Y. Lee, C. Greenberg, E. Godard, A. A. Butt, E. Singer, T. Nguyen, L. Mason, and D. Reynolds, "The 2022 NIST Language Recognition Evaluation," 2023. [Online]. Available: https://arxiv.org/abs/2302.14624

[4] J. Valk and T. Alumäe, "VoxLingua107: a dataset for spoken language recognition," in *IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2021, pp. 652–658.

[5] Y. Lee, C. Greenberg, L. Mason, and E. Singer, "NIST 2022 Language Recognition Evaluation Plan," 2022-08-31 04:08:00 2022, https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=935161.

[6] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, "The Kaldi speech recognition toolkit," in *ASRU*. IEEE Signal Processing Society, 2011.

[7] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust DNN embeddings for speaker recognition," in *ICASSP*. IEEE, 2018, pp. 5329–5333.

[8] D. Snyder, D. Garcia-Romero, A. McCree, G. Sell, D. Povey, and S. Khudanpur, "Spoken Language Recognition using X-vectors," in *The Speaker and Language Recognition Workshop (Odyssey 2018)*, 2018, pp. 105–111.

[9] D. Snyder, G. Chen, and D. Povey, "MUSAN: A Music, Speech, and Noise Corpus," 2015, arXiv:1510.08484v1.

[10] T. Ko, V. Peddinti, D. Povey, M. L. Seltzer, and S. Khudanpur, "A study on data augmentation of reverberant speech for robust speech recognition," in *ICASSP*, 2017.

[11] W. Cai, J. Chen, and M. Li, "Exploring the encoding layer and loss function in end-to-end speaker and language recognition system," in *Odyssey 2018*, 2018.

[12] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016, pp. 770–778.

[13] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *CVPR*, 2018, pp. 7132–7141.

[14] J. Zhou, T. Jiang, Z. Li, L. Li, and Q. Hong, "Deep speaker embedding extraction with channel-wise feature responses and additive supervision softmax loss function," in *Interspeech*, 2019.

[15] Y. Zhu, T. Ko, D. Snyder, B. Mak, and D. Povey, "Self-attentive speaker embeddings for text-independent speaker verification," in *Interspeech*, 2018.

[16] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *Advances in Neural Information Processing Systems*, vol. 33, pp. 12 449–12 460, 2020.

[17] S. Scanzio, P. Laface, L. Fissore, R. Gemello, and F. Mana, "On the use of a multilingual neural network front-end," in *Interspeech*, 2008.

[18] T. Alumäe and J. Valk, "The TalTech systems for the Short-Duration Speaker Verification Challenge 2020," in *Interspeech*, 2020.

[19] S. Shon, A. Ali, Y. Samih, H. Mubarak, and J. Glass, "ADI17: A fine-grained Arabic dialect identification dataset," in *ICASSP*, 2020, pp. 8244–8248.

[20] H. Bredin, R. Yin, J. M. Coria, G. Gelly, P. Korshunov, M. Lavechin, D. Fustes, H. Titeux, W. Bouaziz, and M.-P. Gill, "pyannote.audio: neural building blocks for speaker diarization," in *ICASSP*, Barcelona, Spain, May 2020.

[21] H. Bredin and A. Laurent, "End-to-end speaker segmentation for overlap-aware resegmentation," in *Interspeech*, Brno, Czech Republic, August 2021.

[22] 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," in *Interpseech*, 2022.

[23] D. Povey, G. Cheng, Y. Wang, K. Li, H. Xu, M. Yarmohammadi, and S. Khudanpur, "Semi-Orthogonal Low-Rank Matrix Factorization for Deep Neural Networks," in *Interspeech*, 2018, pp. 3743–3747.

[24] L. F. Lamel and J.-L. Gauvain, "Language identification using phone-based acoustic likelihoods," in *ICASSP*, vol. 1. IEEE, 1994, pp. I–293.

[25] M. A. Zissman, "Comparison of four approaches to automatic language identification of telephone speech," *IEEE Transactions on Speech and Audio Processing*, vol. 4, no. 1, p. 31, 1996.

[26] J.-L. Gauvain, A. Messaoudi, and H. Schwenk, "Language recognition using phone lattices," in *Interspeech*, 2004.

[27] Q. Xu, A. Baevski, and M. Auli, "Simple and effective zero-shot cross-lingual phoneme recognition," in *Interspeech*, 2022.

[28] A. Conneau, A. Baevski, R. Collobert, A. Mohamed, and M. Auli, "Unsupervised Cross-Lingual Representation Learning for Speech Recognition," in *Interspeech*, 2021, pp. 2426–2430.