INTERSPEECH 2023
20-24 August 2023, Dublin, Ireland

Description and Analysis of ABC Submission to NIST LRE 2022

Pavel Matějka[1,2], *Anna Silnova*[1], *Josef Slavíček*[2], *Ladislav Mošner*[1], *Oldřich Plchot*[1], *Michal Klčo*[2], *Junyi Peng*[1], *Themos Stafylakis*[3] *Lukáš Burget*[1],

[1] Brno University of Technology, Speech@FIT, Czech Republic
[2] Phonexia, Czech Republic
[3] Omilia, Greece

matejkap@vut.cz, josef.slavicek@phonexia.com, tstafylakis@omilia.com

Abstract

This paper summarizes our efforts in the NIST Language Recognition Evaluations 2022 resulting in systems providing competitive performance. We provide both the description and analysis of the systems. We describe what data we have used to train our models, and we follow with embedding extractors and backend classifiers. After covering the architecture, we concentrate on post-evaluation analysis. We compare different topologies of DNN, different backend classifiers, and the impact of the data used to train them. We also report results with XLS-R pre-trained models. We present the performance of the systems in the Fixed condition, where participants are required to use only predefined data sets, and also in the Open condition allowing to use any data to train the systems.

Index Terms: language detection, language recognition, embedding extractor, LRE, NIST

1. Introduction

It has been five years since the previous NIST LRE in 2017 [1, 2], where we have seen the dominance of systems based on i-vectors [2, 3] with multilingual bottleneck features [4, 5, 6]. Later in 2018, the trend in language identification followed the speaker recognition field [7, 8, 9] with systems based on DNN embeddings [10, 11, 12].

NIST LRE 2022 defined two conditions - fixed and open. In the fixed condition, only data allowed by NIST could be used so that all systems could be compared from the technology point of view. For this condition, most systems were based on DNN embedding extractors. The second condition is open, where any data could be used. Here, the best systems used embedding extractors built on top of self-supervised pre-trained models [13].

In our submission to NIST LRE2022, we used DNN embedding extractors with different topologies and compared them to XLS-R pre-trained models [13]. We provide a comprehensive description of the systems, calibration, and fusion followed by results.

The analysis part of the paper compares the systems from NIST LRE 2017 and 2022, the performance of several DNN embedding extractors, shows the superiority of multilingual pre-trained XLS-R models and proves the importance of the development set with data from the target domain. For the fixed condition, it was the first evaluation where we did not have the data from target languages available for the embedding extractor training, therefore, it was more similar to the open-set style training, where we have to build a general embedding extractor.

The work was supported by Czech Ministry of Interior project No. VJ01010108 "ROZKAZ", Czech National Science Foundation (GACR) project NEUREM3 No. 19-26934X, Czech Ministry of Culture NAKI III project JARIN (DH23P03OVV010), Czech Ministry of Education, Youth and Sports project no. LTAIN19087 "Multi-linguality in speech technologies", and Horizon 2020 Marie Sklodowska-Curie grant ESPERANTO, No. 101007666. Computing on IT4I supercomputer was supported by the Czech Ministry of Education, Youth and Sports through the e-INFRA CZ (ID:90140).

2. Data

The results and analysis in this paper are linked to the 2022 NIST language recognition evaluation (LRE22), the 9th edition of an ongoing language recognition evaluation series that started in 1996.

The **LRE22 eval set** contains 14 target languages [14]. The emphasis is on African languages, including low-resource ones. The duration of the files is randomly sampled between 5 and 30 sec. The sampling frequency is 8kHz. The data were collected from 2 different sources (Conversational Telephone Speech (CTS) and Broadcast Narrow Band Speech (BNBS)) and from 3 different collections, recorded outside of North America [14].

The **LRE22 dev set** is a small development set with the same properties as the LRE22 eval set. All speech files are cuts from 30 recordings per target language, where each recording comes from a unique speaker. This set was the only evidence of the data for the target languages, and we used it to train the classification backend. For development, we split it into 2 halves (each containing 15 unique speakers per language) and used one for training and the other for performance monitoring.

For the **training data in Fixed condition**, we utilized all data supplied by NIST (training and development).The only allowed data according to the evaluation plan [14] were training, development, and test data from NIST LRE 2017 (14 languages) and VoxLingua107 (a set of 107 languages [15]).

Training data in Open condition allowed using any data. We have constructed 2 sets of corpora for training or fine-tuning. The details and statistics of are in Table 1.

Set 1 contains data from: KALAKA [16], OGI multi-language [17], OGI 22 languages [17], ADI17 [18], ELRA SpeechDat(E) [19], MSIL18 [20], Radio Free Europe [21] and data from projects - LORELEI [22], MATERIAL [23], WELCOME [24].

Set 2 contains data from: Mozilla CommonVoice [25], Private Phonexia call-center data (1100 hours, 87k utterances, 13 languages), and LDC dataset [17] - QT Levantine Arabic, Gulf Arabic CTS, GALE Arabic Broadcast, GALE Chinese Broadcast, RuSTeN (Russian), CIEMPIESS (Spanish)

Both sets contain these datasets: VoxLingua107 [15], LWAZI [26], Voice of America [27], BABEL project data [17], and the data from LDC [17] - Callfriend, Callhome, FAE, Fisher English and Levantine Arabic, HKUST Mandarin, NIST LRE & SRE data.

We constructed **BUT dev set** because the LRE 2022 development set contained only the 30 unique recording sessions.

10.21437/Interspeech.2023-1529

Table 1: *Statistics of training data.*

Database	Closed Set		Open Set	
	NIST LRE 2017	VoxLingua	Set 1	Set 2
# languages	14	107	106	135
# files	45k	2537k	730k	2864k
# hours	1013	5509	9730	14764

We, therefore, decided to create our own set to train the backend for the open condition. This set contains 14 target languages coming from several sources: LWAZI [26], ADI17 [18], LORELEI [22], NIST LRE 2015 [28], NIST SRE 2017 [1] and data from Voice of America [27]. It has 8491 recordings, 34 hours of speech after energy-based VAD. For most of the languages, we have between 20 to 120 minutes of data.

3. Systems

All our systems include an embedding extractor paradigm trained on data respecting the target condition (fixed or open) and a backend (Gaussian Linear Classifier (GLC) or Probabilistic Linear Discriminant Analysis (PLDA)) trained on the LRE22 dev data. This section describes the individual systems listed in Table 2. The first part of the table shows the results of individual systems and fusion for the fixed condition. The second part is dedicated to the open condition. In each system, the DNN-based embedding extractor is trained to discriminate between languages in the training set (e.g. using AAM loss), where the number of training languages differs for the fixed condition and open condition. We provide high-level information in this section and focus more on the next experiment section. See [29] for more information about how the systems were trained and detailed parameter settings.

The input to all systems is 8kHz speech (downsampled if necessary). The only exceptions are systems based on XLS-R, which is pre-trained on 16kHz data. For these systems, we resampled training data to 16kHz.

During the development, we used only half of the LRE22 dev set for training the backend and calibration parameters. The second half was used to track the performance. However, for the final submission, the systems were retrained on the whole LRE22 dev set. The effect on the performance is reported in Table 2 and described in Section 4.

The task for LRE 2022 is to provide an output in the form of a vector of log-likelihood scores, one score per target language which is a task of a backend classifier (GLC, PLDA). The evaluation metric – the cost $C_{primary}$ – is given by LRE evaluation plan [14]. Note that $C_{primary}$ is a calibration-sensitive metric; thus, we report both the cost with the analytically set threshold minimizing the fusions risk (actC) and the minimum of the cost achieved by setting the threshold in an oracle way (minC).

If not stated otherwise, we used energy-based VAD. Our VAD does not use any training data and, therefore, can be used for the fixed condition. We train 1D Gaussian Mixture model (GMM) in an unsupervised fashion for each test file on the sequence of frames energies. The GMM has 3 Gaussian components. We then discard the frames assigned to the Gaussian representing the lowest energy frames in the recording.

3.1. DNN embeddings

3.1.1. Fixed condition – ResNet34-CE, ResNet34-AAM, ResNet34-2head

All models are based on the same backbone architecture derived from the standard computer vision model [30]. As opposed to the original ResNet34, the first convolution has a kernel size of 3 and a stride of 1. Statistics are projected to 256-dimensional embeddings.

The models differ in the training objective. ResNet34-CE was trained with a softmax (cross-entropy) loss, whereas, for ResNet34-AAM, we used additive angular margin loss (AAM) [31] with a margin set to 0 and a scale to 30. Both ResNet34-CE and ResNet34-AAM were trained on the VoxLingua107 dataset [15]. We did not apply VAD for VoxLingua107 corpus. Development and evaluation data of the challenge contain fine-grained labels (at the level of dialects), which contradicts the coarse-grained labels of VoxLingua107 (e.g., for English and Arabic languages). To allow the ResNet34-AAM network to learn more than one representation per VoxLingua107 languages, we adopted sub-center ArcFace [32] with 3 sub-centers.

ResNet34-2head optimized the same AAM loss as ResNet34-AAM and also utilized 3 sub-centers for each class. Contrary to previous systems, it was trained on VoxLingua107 and NIST LRE 2017 datasets. They contain labels of different granularity (e.g. VoxLingua107 has one class Arabic while NIST LRE 2017 distinguishes 4 of Arabic dialects.). To tackle this issue, the model has two classification heads (one for each corpus). Head and example correspondence to the dataset is taken into account when evaluating the loss.

3.1.2. Fixed condition – ResNet100

The ResNet100 system was inspired by the speaker ID system from [33] with a different number of output channels for each stage (32, 64, 128, 256). In each ResNet block, the frequency-wise Sqeeze-Excitation (fwSE) [34] block was incorporated with a bottleneck size of 128. Frame-by-frame ResNet outputs are aggregated with statistics pooling and projected to 256-dimensional embedding. We applied data augmentation during training using MUSAN [35] and RIR [36] corpora.

The system was trained for 3 epochs on Voxlingua107 with AM-softmax loss[1]. We set the margin to 0 and the scale to 32. The learning rate was scheduled in the same way as in [33].

3.1.3. Fixed condition – ECAPA-TDNN

As an alternative to ResNet systems, we trained also ECAPA-TDNN [37] where we followed the recipe from SpeechBrain[2]. The model contains approximately 14M parameters and is trained with AAM loss and the margin set to 0 to discriminate between the languages in the training set for fixed conditions.

3.1.4. Fixed condition – RepVGG

The system based on the RepVGG architecture utilizes the B1 variant from [38]. The training setup and parameters are the same as in the training of ResNet100 except for batch size, which is 380.

3.1.5. Open condition – ResNet101

The ResNet101 model is, in essence, a scaled-up version of ResNet34 networks. Due to increased GPU memory requirements, each minibatch contained 32 examples. Input feature extraction is equivalent to that for ResNet34 models. The model was trained using the training data Set 1 described in Section 2 and Table 1 with the AAM loss with a margin of 0 and a scale of 30.

3.1.6. Open condition – XLSR Models

For the open condition, we used two systems based on pre-trained XLS-R-1B [13], downloaded from HuggingFace[3]. We finetune the systems as language classifiers with TDNN. We

[1] However, with margin 0, AAM and AM-softmax loss are the same.

[2] `https://github.com/speechbrain/speechbrain/`

[3] https://huggingface.co/facebook/wav2vec2-xls-r-1b

adopted TDNN architecture from [39]. The systems are fine-tuned with the softmax loss with Adam [40] optimizer. The only difference between our two XLS-R systems is the learning rate used in training. The systems ware trained on Set 2 described in Section 2 and Table 1. The details of the training procedure, together with parameter settings, are reported in [anonymized].

For embedding extraction, we upsample the data to 16kHz and process them with a sliding window of maximal length 20s. We scan the whole utterance by shifting the window by 2s and then average embeddings obtained from all windows.

3.2. Classifiers

The NIST LRE 2022 protocol required that the participants submit their system outputs in the form of log-likelihood scores, one score per target language. For the majority of our systems, the log-likelihood scores were evaluated using Gaussian Linear Classifier (GLC) [41]. This model assumes that the DNN embeddings for each language follow Gaussian distribution. Each Gaussian is assumed to have a language-dependent mean vector, and fixed covariance matrix shared across all languages. The parameters of GLC are Maximum-Likelihood estimated on the development data. In all cases, we use half of the LRE22 dev set to estimate the GLC parameters. For ResNet34 systems from the fixed condition and ResNet101 from the open condition, the training data were expanded by including the embeddings extracted from the augmented version of the LRE22 dev set (one augmented utterance per one original). Finally, for ResNet101 embeddings, we trained a second GLC backend (see line 12 in Table 2) on the augmented version of the LRE22 dev set and BUTdev utterances.

Unlike all the other systems, the embeddings extracted from the XLS-R-1b (3e-6) model were modeled using a Probabilistic Linear Discriminant Analysis (PLDA) model [42]. In this case, we train the PLDA on half of the LRE22 dev set using language labels. Then, to generate the vector of scores, each of the test utterances is scored against 14 "enrolled models". Each enrolled model is composed of training utterances belonging to the same language. We average the embeddings from the enrollment segments sharing the same session id (i.e., are cuts from one original LRE22 dev recording of one speaker). The final score (playing the role of the required log-likelihood score) for each language is the log-likelihood ratio (LLR) score for the multi-enroll/single-test verification trial, with the number of enrollment segments equal to the number of sessions used from a given language (i.e. 15 when half of the LRE22 dev set is used).

For most of the systems, before training the back-end (GLC or PLDA), we reduce the dimensionality of the embeddings by Principal Component Analysis (PCA); the respective target PCA dimensionalities are reported in Table 2. For some of the embeddings, the target dimensionality after PCA was selected automatically: there was an obvious gap in the eigenvalues of the covariance matrix estimated on the training data, i.e., some dimensions had very little variability compared to the others.

For the embeddings where we did not observe a similar pattern (that is, if the eigenvalues were "smoothly" increasing when sorted), we experimented with several options for PCA dimensionality and selected the one performing the best on the held-out development set.

3.3. Calibration and fusion

After retrieving the scores from the individual systems, we proceeded with two additional steps: calibration and fusion.

We utilize Logistic Regression (LR) calibration: a single scalar and a vector of offsets (one per language) are learned by optimizing the cross-entropy objective. During model development, we trained the calibration parameters on half of the LRE22 dev set and tested the performance on the held-out half of the development set. The calibration parameters are learned on the same data that were used to train the backend classifier. When training the final model for the submission, we used the whole LRE22 dev set for both backend and calibration training.

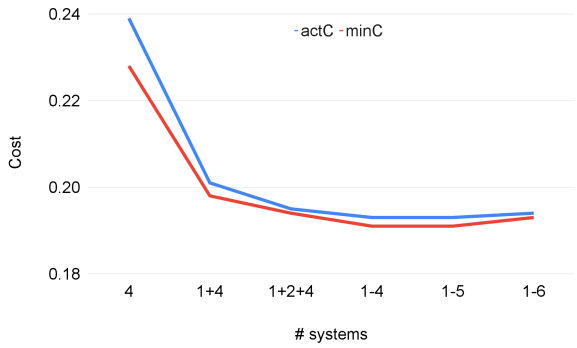

Figure 1: *Trend of the performance of different numbers of systems in the fusion on the evaluation data.*

The calibration stage was applied to all of the systems submitted to the fixed condition and to two systems eligible for the open condition: SBN i-vector [3] and one of ResNet101. The remaining three systems submitted to the open condition were not calibrated due to the low amount of errors that they had on our calibration set.

In all cases, the fusion step did not require training any parameters and consisted of averaging the (calibrated) scores from the individual systems.

4. Experiments and analysis

In this section, we analyze and comment on numerous experiments that we performed with our systems during LRE22 dev and in the post-evaluation period. Primarily, we report results on the LRE22 evaluation dataset.

Our primary submission to fixed condition was a fusion of 4 systems. Lines 1 to 4 in Table 2 show separate systems fused to Primary submission (line 7) with $minC = 0.191$. There is a 20% relative gain from the fusion compared to the single best system. Contrastive submission (line 8) is a fusion of 6 systems (line 1 to 6). Performance of the fusion as a function of number of fused systems (Figure 1) shows that most of the gain comes from the first two systems. The single best system is a ResNet34-2head with $minC = 0.228$. Using 2 heads (one trained on Voxlingua and the second on NIST LRE2017 data) brings 25% relative improvement over the ResNet34-AAM.

The second half of Table 2 shows the results in the Open condition. Here, the primary submission (line 14) is a fusion of the two XLS-R models with different learning rates and backends (lines 9 and 10) and one ResNet101 (line 11). We see a 15% relative gain compared to the single best system.

For comparison, we also report the performance of our single best system (ivectors extracted from multilingual bottleneck features) that was submitted to the previous NIST LRE 2017 (line 13 from Table 2). We can see 70% relative improvement brought by the single best XLS-R system reported on line 10 in Table 2.

In Table 2, we can see that all systems perform better on the LRE22 eval set than on the LRE22 dev set. The main reason for this is that the backend is trained only on half of the LRE22

Table 2: *Performance of the submitted systems on half of the LRE 2022 development set and on the evaluation set. The systems marked with "*" are submitted as contrastive single systems.*

		Embeddings	Back-end	Calibration	½ LRE DEV 22		LRE 22 EVAL	
					minC	actC	minC	actC
Fixed condition	1*	ResNet100	PCA 117, GLC	LR	0.294	0.314	0.250	0.256
	2	RepVGG	PCA 117, GLC	LR	0.340	0.373	0.266	0.275
	3	ResNet34-CE	PCA 100, GLC	LR	0.330	0.354	0.268	0.275
	4	ResNet34-2head	GLC	LR	0.274	0.296	0.228	0.239
	5	ResNet34-AAM	PCA 100, GLC	LR	0.313	0.329	0.283	0.284
	6	ECAPA-TDNN	PCA 100, GLC	LR	0.409	0.458	0.342	0.354
	7	Primary submission 1+2+3+4			0.235	0.243	0.191	0.193
	8	Contrastive submission 1+2+3+4+5+6			0.236	0.244	0.193	0.194
Open condition	9*	XLS-R-1b LR=1e-6	PCA 100, GLC	-	0.113	0.123	0.102	0.110
	10	XLS-R-1b LR=3e-6	PCA 300, PLDA	-	0.114	0.130	0.095	0.104
	11	ResNet101	PCA 151, GLC	-	0.167	0.182	0.152	0.163
	12*	ResNet101	PCA 151, GLC(+BUTdev)	LR	0.172	0.181	0.148	0.152
	13*	LRE 2017 BUT BabelSBN ivectors [3]	PCA 100, GLC	LR	0.379	0.392	0.307	0.310
	14	Primary submission 9+10+11			0.097	0.108	0.082	0.088

Table 3: *Analysis of the backend training set for system ResNet34-AAM with PCA100 and GLC backend without calibration. Results are on the NIST LRE 2022 evaluation data. For systems using LRE17, the GLC covariance matrix is an average between covariances estimated on LRE17 and LRE22 dev.*

Backend training set	minC	actC
½ LRE22 dev	0.358	0.403
½ LRE22 dev + augs	0.342	0.372
½ LRE22 dev + LRE17	0.341	0.376
LRE22 dev	0.304	0.325
LRE22 dev + augs	0.297	0.310
LRE22 dev + augs + LRE17	0.297	0.308

dev set, as we report the performance on the other half. Table 3 shows that about 20% relative degradation in performance is observed also for LRE22 eval set when training the GLC backend on only half of the LRE22 dev set.

Table 3 also analyzes two strategies for obtaining more robust estimates of the GLC backend parameters. The first strategy is to train the GLC also on the augmented LRE22 dev set. The within-class covariance matrix accounts for most of the parameters representing the GLC backend. Therefore, the second strategy is to obtain a robust estimate of the covariance matrix by interpolating the one estimated on LRE22 dev set with another estimated on many embeddings (including non-target languages) extracted from NIST LRE 2017 data. The best option is to use the whole NIST LRE22 dev set with one set of different augmentations.

Table 4 reports the results of our effort with XLS-R systems. The change in learning rate in training (first 2 rows) does not have much effect on the performance. By changing the classifier from GLC to PLDA, we see a small improvement for a single system. The most gain, however, comes from the fusion of these two changes. Fusing systems where we have different learning rates and different backends (line 5) has 10% relative improvement over the single system (line 3). Fusing the system with different learning rates and the same backend yield only 4% relative improvement.

Table 4: *Analysis of calibration for open condition. Results are on the NIST LRE 2022 evaluation data.*

#	System	minC	actC
1	XLS-R-1b LR=1e-6, GLC	0.102	0.110
2	XLS-R-1b LR=3e-6, GLC	0.103	0.114
3	XLS-R-1b LR=3e-6, PLDA	0.095	0.104
4	avg fusion (1+2)	0.091	0.098
5	avg fusion (1+3)	0.085	0.092

Table 5: *Effect of not having the NIST LRE 2022 development set. The results are with ResNet 101 on open condition and embedding reduction with PCA to 151 on the LRE22 eval set.*

Backend	LRE2022 eval set	
ResNet 101, PCA151	minC	actC
GLC(LRE22dev)	0.152	0.163
GLC(LRE22dev+BUT dev)	0.148	0.152
GLC(BUT dev)	0.736	0.836

During the post-analysis, we tried to answer the question from the LRE22 workshop - "Do we need a small portion of target domain data for system development?" We constructed BUT dev set from the available data given the target languages. We excluded this data from the embedding extractor training and trained a new ResNet101 system for open condition.

Table 5 compares the systems where the GLC backend is trained only on target domain LRE22 dev set (first row), only on the BUT dev set without the target domain data (last row) or on both (middle row). The system that has not seen the target domain data is 5 times worse than the one which has seen this data. These results show that for good performance, we need to see at least a small portion of the target domain data in the backend. The results also suggest that our embedding extractor is not robust enough against channel, or the data we have composed in BUT dev set is too far from the target domain.

5. Conclusion

In this paper, we summarized our efforts for the NIST LRE 2022. We trained the systems for fixed and closed conditions. This evaluation showed the superiority of pre-trained models such as XLS-R, we obtained more than 30% relative gain over the state-of-the-art ResNet embeddings. On the LRE22 task (open condition), our best single system developed for LRE22 shows a 70% relative improvement over our best single system developed for LRE17. The official results of all submitted systems to NIST LRE 2022 evaluation are published in [43].

6. References

[1] "NIST 2017 Language Recognition Evaluation Plan," https://www.nist.gov/sites/default/files/documents/2017/06/01/lre17_eval_plan-2017-05-31_v2.pdf.

[2] Seyed Omid Sadjadi et al, "Performance Analysis of the 2017 NIST Language Recognition Evaluation," in *Proc. Interspeech*, 2018.

[3] O. Plchot and et al., "Analysis of BUT-PT submission for NIST LRE 2017," in *Proceedings of Odyssey 2018 The Speaker and Language Recognition Workshop*, 2018.

[4] R. Fer, P. Matejka, F. Grezl, O. Plchot, K. Vesely, and J. H. Cernocky, "Multilingually trained bottleneck features in spoken language recognition," *Computer Speech & Language*, vol. 46, no. Supplement C, pp. 252 – 267, 2017.

[5] Y. Song, B. Jiang, Y. Bao, S. Wei, and L.-R. Dai, "i-vector representation based on bottleneck features for language identification," *Electronics Letters*, vol. 49, no. 24, pp. 1569–1570, 2013.

[6] P. Matějka, L. Zhang, T. Ng, H. S. Mallidi, O. Glembek, J. Ma, and B. Zhang, "Neural network bottleneck features for language identification," in *Proceedings of Odyssey 2014*, 2014.

[7] E. Variani, X. Lei, E. McDermott, I. L. Moreno, and J. Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2014, pp. 4052–4056.

[8] D. Snyder, D. Garcia-Romero, D. Povey, and S. Khudanpur, "Deep Neural Network Embeddings for Text-Independent Speaker Verification," in *Proceedings of Interspeech*, 2017.

[9] G. Bhattacharya, J. Alam, and P. Kenny, "Deep Speaker Embeddings for Short-Duration Speaker Verification," in *Interspeech 2017*, 08 2017, pp. 1517–1521.

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

[11] A. Lozano-Diez, O. Plchot, P. Matejka, and J. Gonzalez-Rodriguez, "DNN Based Embeddings for Language Recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018.

[12] A. D. Lozano, O. Plchot, P. Matějka, O. Novotný, and J. Gonzalez-Rodriguez, "Analysis of DNN-based embeddings for language recognition on the NIST LRE 2017," in *Proceedings of Odyssey 2018 The Speaker and Language Recognition Workshop*.

[13] 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," *CoRR*, 2021.

[14] Y. Lee, C. Greenberg, L. Mason, and E. Singer, "The 2022 NIST language recognition evaluation plan LRE22." [Online]. Available: https://www.nist.gov/publications/nist-2022-language-recognition-evaluation-plan

[15] J. Valk and T. Alumäe, "Voxlingua107: A dataset for spoken language recognition," *IEEE Spoken Language Technology Workshop (SLT)*, 2020.

[16] L. J. Rodríguez-Fuentes, M. Penagarikano, G. Bordel, A. Varona, and M. Díez, "KALAKA: A TV broadcast speech database for the evaluation of language recognition systems," in *Proceedings of LREC*, May 2010.

[17] "Linguistic Data Consorcium LDC catalog," https://catalog.ldc.upenn.edu/.

[18] S. Shon, , A. Ali, Y. Samih, H. Mubarak, and J. Glass, "ADI17: A fine-grained Arabic dialect identification dataset," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 8244–8248.

[19] "ELRA catalog," http://catalog.elra.info/.

[20] B. M. L. Srivastava, S. Sitaram, R. K. Mehta, K. D. Mohan, P. Matani, S. Satpal, K. Bali, R. Srikanth, and N. Nayak, "Interspeech 2018 Low Resource Automatic Speech Recognition Challenge for Indian Languages," in *Workshop on Spoken Language Technologies for Under-resourced Languages*, 2018.

[21] "Radio Free Europe," www.rferl.org.

[22] S. Strassel and J. Tracey, "Lorelei language packs: Data, tools, and resources for technology development in low resource languages," in *International Conference on Language Resources and Evaluation*, 2016.

[23] "IARPA project MATERIAL," https://www.iarpa.gov/index.php/research-programs/material.

[24] "EU project WELCOME," https://welcome-h2020.eu/.

[25] "Mozilla Common Voice," https://commonvoice.mozilla.org/.

[26] C. van Heerden, N. Kleynhans, and M. Davel, "Improving the LWAZI ASR baseline," in *Interspeech*, 2016.

[27] "Voice of America," www.voanews.com.

[28] "The 2015 NIST Language Recognition Evaluation Plan (LRE15)," http://www.nist.gov/itl/iad/mig/upload/LRE15_EvalPlan_v23.pdf.

[29] "ABC System Description for NIST LRE 2022." [Online]. Available: https://www.fit.vut.cz/research/publication/12986/

[30] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[31] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "ArcFace: Additive Angular Margin Loss for Deep Face Recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[32] J. Deng, J. Guo, T. Liu, M. Gong, and S. Zafeiriou, "Sub-center ArcFace: Boosting Face Recognition by Large-Scale Noisy Web Faces," in *Computer Vision – ECCV 2020*, 2020, pp. 741–757.

[33] R. Makarov, N. Torgashov, A. Alenin, I. Yakovlev, and A. Okhotnikov, "ID R&D system description to VoxCeleb speaker recognition challenge 2022," *ID R&D Inc.: New York, NY, USA*.

[34] J. Thienpondt, B. Desplanques, and K. Demuynck, "Integrating frequency translational invariance in TDNNs and frequency positional information in 2d ResNets to enhance speaker verification," *INTERSPEECH*, 2021.

[35] D. Snyder, G. Chen, and D. Povey, "MUSAN: A music, speech, and noise corpus," *arXiv preprint arXiv:1510.08484*, 2015.

[36] 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.

[37] B. Desplanques, J. Thienpondt, and K. Demuynck, "ECAPA-TDNN: Emphasized Channel Attention, Propagation and Aggregation in TDNN Based Speaker Verification," in *Interspeech*, 2020.

[38] M. Zhao, Y. Ma, M. Liu, and M. Xu, "The SpeakIn system for VoxCeleb speaker recognition challange 2021," *arXiv preprint arXiv:2109.01989*, 2021.

[39] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust DNN Embeddings for Speaker Recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018.

[40] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014.

[41] D. Martínez, O. Plchot, L. Burget, O. Glembek, and P. Matějka, "Language Recognition in iVectors Space," in *Proc. of Interspeech*, 2011.

[42] P. Kenny, "Bayesian speaker verification with Heavy–Tailed Priors," in *keynote presentation, Odyssey 2010*, 2010. [Online]. Available: https://www.superlectures.com/odyssey/lecture.php?lang=en&id=15

[43] Y. Lee, C. Greenberg, E. Godard, A. A. Butt, E. S. andTrang Nguyen, L. Mason, and D. Reynolds, "The 2022 NIST Language Recognition Evaluation," in *Interspeech*, 2022. [Online]. Available: https://arxiv.org/pdf/2302.14624.pdf