INTERSPEECH 2023
20-24 August 2023, Dublin, Ireland

End to End Spoken Language Diarization with Wav2vec Embeddings

Jagabandhu Mishra[1], *Jayadev N Patil*[2], *Amartya Chowdhury*[1], *S. R. Mahadeva Prasanna*[1]

[1]Department of Electrical Engineering, Indian Institute of Technology (IIT) Dharwad, India
[2]Department of Electronics and Communication Engineering, KLE Technological University, India

jagabandhu.mishra.18@iitdh.ac.in, jayadevnpatil@gmail.com, {amartya.chowdhury, prasanna}@iitdh.ac.in

Abstract

The performance of the available end-to-end (E2E) spoken language diarization (LD) systems is biased towards primary language. This is due to the unavailability of sufficient secondary language data in code-switched (CS) utterances. Hence, to resolve the issue, this work initially uses wav2vec (W2V) pre-trained embeddings in place of x-vector to reduce the primary language bias and provides a relative improvement of 30.7% in terms of Jaccard error rate (JER) over the baseline x-vector based E2E (X-E2E) framework. Further, the performance of LD is improved by fine-tuning the W2V embeddings extractor and modifying the temporal aggregation strategy from statistical pooling to attention pooling. The Final performance achieved in terms of JER is 21.8, which provides a relative improvement of 40.7% and 63.9% over the standalone W2V fine-tuned and the baseline X-E2E framework, respectively.

Index Terms: Spoken language diarization. wav2vec, Language data imbalance

1. Introduction

Spoken language diarization (LD) aims to automatically segment and label the monolingual segments present in a given code-switched (CS) utterance. The increasing demand for the deployment of speech applications in multilingual CS scenarios motivates the development of the LD system [1, 2, 3, 4, 5, 6, 7, 8]. The success of E2E frameworks in speaker diarization (SD) inspires the development of LD systems using E2E framework [5, 9, 10]. In [5], an E2E x-vector framework was presented to perform the LD task. Apart from this, the available attempts towards the development of LD systems are limited in the literature. With the Microsoft code-switch (MSCS) data, some attempts are made that use deepspeech2 (DS2) [11], transformer [12], DS2 with secondary language masking (DS2-LM) [13] and wav2vec (W2V) [14], for performing sub utterance level language identification (SLID) task. SLID predicts language tags for each fixed duration segment of a given utterance, hence can be considered as the first-level LD system.

Generally, in code-switched utterances, the turn duration of the primary language is significantly larger than the secondary language. The same can be observed from the plot depicted in Figure 1(b). For all three language pairs of the MSCS data, the average turn duration of the primary language is around 1.5 seconds. Whereas, for the secondary language, the duration is 0.5 seconds. The turn imbalance will lead to data imbalance in the training set. Figure 1(a) shows the percentage of the primary and secondary language duration in the training set of the MSCS corpus. It can be observed from the figure that for three language pairs, the primary language percentage is more than 80%. Further, in the CS utterances, the available primary and

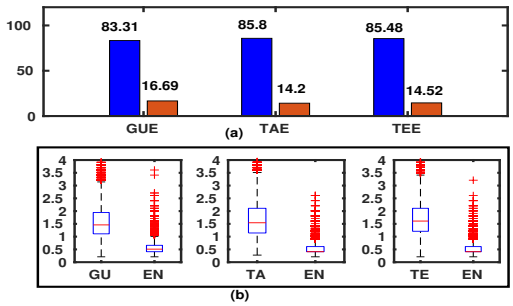

Figure 1: *MSCS train set: (a) data duration (in %), (b) monolingual segment distribution (in seconds).*

secondary languages are spoken by a single speaker. On the other hand, the secondary language is generally produced by adapting the language production system of the primary language may lead to acoustic similarity. In such a scenario, as most LD and SLID frameworks use a discriminative strategy to train the neural network framework, the imbalance in data and acoustic similarity may bias the system performance towards the primary language.

In [15] and [14], the issue of primary language bias of the DS2 and DS2-LM-based framework is discussed, and reduced the effect by using W2V embeddings. The works also mentioned that most of the LD and SLID frameworks use identification accuracy (IDA) and equal error rate (EER) as performance measures. Due to the turn imbalance in test utterances, even if the system performance is biased towards one class, the IDA and EER mislead the performance interpretation by providing high and low values, respectively. Inspired by SD literature this work suggests the use of Jaccard error rate (JER) for the calibration of LD systems while dealing with the utterances having turn imbalance [14]. In [15], the W2V embedding extractor was pre-trained with approximately 1000 hours of data from 23 Indian languages. The pre-training stage uses contrastive divergence loss to optimize the parameters of the framework by predicting the embedding vectors for the masked portion of an utterance [16, 17]. A minimum masking duration of 320 ms was used and hypothesized that the trained network can predict the embedding corresponds to syllables/sub-words independent of the language [18, 14]. Therefore, by employing the appropriate fine-tuning strategy, even with a small amount of data we may be able to discriminate between the languages. Conversely, the x-vector-based E2E system may end up providing biased performance due to imbalanced data and acoustic similarity. Inspired by this fact, this work initially investigates the performance bias of the baseline x-vector-based E2E framework and

10.21437/Interspeech.2023-1109

then explores the W2V embeddings with the E2E framework for improving the performance of LD by reducing the primary language bias.

The rest of the paper is organized as follows: Section 2, discusses the performance bias of the available LD frameworks and also discussed the motivation of the work. The proposed W2V-E2E framework is discussed in Section 3. Section 4, describes the experimental setup and results. Finally, the conclusion and future directions are discussed in Section 5.

2. Motivation of using W2V framework

We have considered the MSCS corpus for our experiments by observing the primary language bias of the corpus and replicating the few available works using the corpus. The MSCS dataset has training and development data from three language pairs: (a) Gujarati-English (GUE), (b) Tamil-English (TAE), and (c) Telugu-English (TEE). For reproducing the results, the architectures and the corresponding hyper-parameters reported in the respective literature are followed here. The obtained IDA and confusion matrix (average across all language pairs) are presented in Table 1.

Table 1: *Comparison between confusion matrix of the available approaches for LD, P: primary, S: secondary, and Sil: silence.*

Model	IDA (%)		P	S	Sil
DS2 [11]	72.2	P	85.2	6.7	8
		S	62.1	**30.4**	7.4
		Sil	31.9	4.5	63.5
DS2 - LM [13]	74.7	P	90.6	1.3	7.9
		S	79.9	**12.3**	7.6
		Sil	29.1	1.3	69.3
x-vector [5]	81.3	P	90.6	0	9.3
		S	65.2	**0**	34.7
		Sil	16.8	0	83.1
W2V [14]	82.0	P	90.9	3.8	5.1
		S	31.3	**63.9**	4.6
		Sil	23	4.6	72.3

It can be observed from the table that DS2, DS2-LM, and x-vector frameworks are providing IDA more than 70%, but their secondary to secondary language identification rate is 30.4%, 12.3% and 0%, respectively. Further, the secondary language segments miss-classified to primary are 62.1%, 79.9%, and 65.23% for DS2, DS2-LM, and x-vector frameworks, respectively. These results show that the system performance is biased towards the primary language. On the other hand, the W2V framework used for the SLID task can identify secondary to secondary language 63.9% and secondary to primary language 31.3%. This shows that the W2V embeddings can reduce the primary language bias to some extent. Hence, motivated by these results, we have done a rigorous exploration of W2V embeddings with the E2E framework to perform the LD task.

3. Proposed W2V based E2E framework

The E2E framework used in this study was originally proposed in [5, 9]. We have replaced the x-vector extraction framework with a W2V extractor. The architecture of W2V and its training strategy used here are borrowed from [17, 14] and [15]. Inspired by the fact that attention pooling (AP) is a better temporal aggregation strategy than statistical pooling (SP) [19], we have performed a comparative analysis using both strategies. The block diagram of the W2V-E2E architecture using SP (W2V-ES) and AP (W2V-EA) is shown in Figure 2 (c) and (d). The W2V pretraining and finetuning strategy is depicted in Figure 2 (a) and (b).

3.1. W2V-E2E architecture

The W2V pre-training and fine-tuning architecture used in [17, 15] are used here without any modification. After performing pre-training and fine-tuning, the "A" block shown in Figure 2(a) and (b) is detached and used as a feature extractor for W2V-ES and W2V-EA. For W2V-ES, the 768 dimensional W2V features are extracted from the speech signal in every 20 ms, and then statistical pooling is performed in every 200 ms. Similarly, For W2V-EA, the W2V features are passed through two linear layers of 768 dimension, and then through attention pooling. After that, the vectors are aggregated in every 200 ms and again passed through the linear layer of dimension 1536.

The 1536 dimension output vectors are further passed through the two linear layers of dimension 3000 and 256 and given input to the classification and self-attention block. The self-attention block consists of layer normalization, positional encoding, another layer normalization followed by J number of encoder layers, and finally a linear layer with sigmoid activation. In this study, the number of self-attention head J is considered as 4. The input to the classification block is batch-normalized first and then passed through a linear layer of dimension 256 and then given input to the softmax layer. In this work, the deep clustering is represented using the self-attention block as mentioned in [9, 5].

3.2. Training strategy

3.2.1. W2V Pre-training

The pre-training W2V architecture shown in Figure 2(a) consists of four operations: (a) feature extraction using convolutional neural networks, (b) quantization using product quantizer, (c) sequence learning using the transformer, and (d) contrastive divergence loss. As shown in the figure the output Z is masked randomly by considering the minimum number of frames (M) as 16 and letting the transformer predict the output C for the masked regions. The actual Z is quantized and the output Q is compared with C using contrastive and divergence loss. As Z is computed in every 20 ms and the minimum M value is 16, it is expected that during training the architecture learns the temporal dynamics to predict syllables/sub-words. The detailed description of the pre-training process can be found at [14, 16]. The pre-training model, which was trained using the 23 Indian language and originally proposed for automatic speech recognition tasks and available in [16]), is used here.

3.2.2. W2V Fine-tuning

In pre-training, it is expected that the network learns to predict the embedding belonging to the syllable/sub-words by capturing the long-term temporal dynamics. In Fine-tuning stage, the objective is to learn the discrimination between the languages. Hence, the label sequence and corresponding speech utterances available in the training set of MSCS are used to fine-tune the W2V framework. As mentioned in Figure 2(b), the "A" block is detached from the pre-training framework and then added with a softmax linear layer and trained using connectionist temporal loss.

3.2.3. E2E training

The pre-trained and fine-tuned model are used to extract features and trained separately using W2V-ES and W2V-EA. For

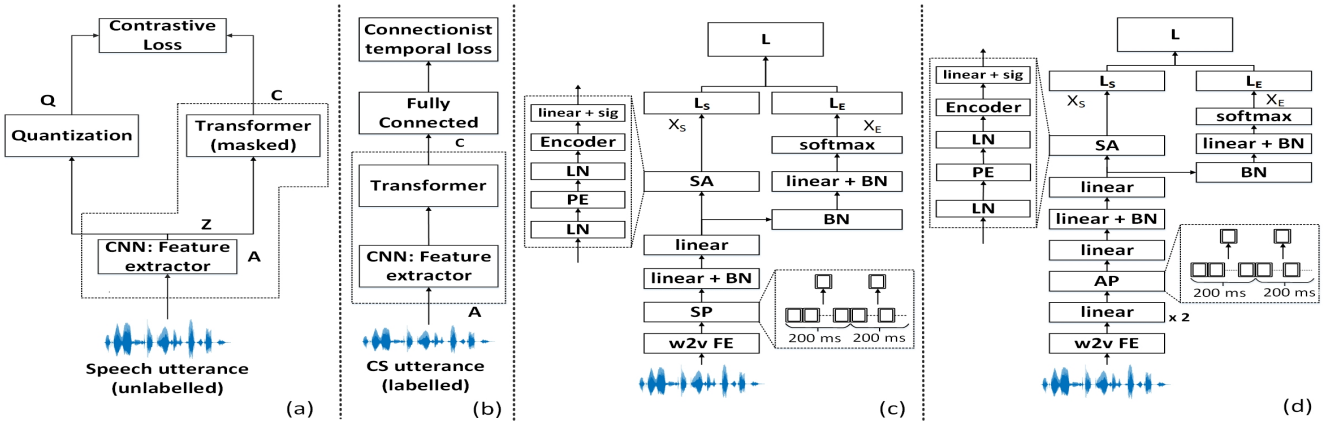

Figure 2: *W2V-E2E architecture (a) W2V-pretraining, (b) W2V-finetuning, (c) W2V-ES (d) W2V-EA, SA: self attention, PE: positional encoding, linear+sig: linear with sigmoid.*

training the E2E framework the joint loss (L) of self-attention (L_S) and classification embedding (L_E) loss is used. The computation of L is given in Eq. 1. The computation of L_E and L_S are defined in Eq 3 and Eq 2, respectively. Where $CE(.)$ represents cross-entropy loss, Y represents ground truth label sequence, and X_S, X_E predicted sequence from self-attention and classification embedding block, respectively.

$$L = \alpha L_S + (1 - \alpha) L_E \tag{1}$$

$$L_S = CE(Y, X_S) \tag{2}$$

$$L_E = CE(Y, X_E) \tag{3}$$

4. Experimental Setup, result, and discussions

4.1. Dataset description

The experiments were conducted using the MSCS dataset released by Microsoft [11]. The sub-utterance language ids are available in each 200 ms duration of each utterance. The training set was used for training while the development set was used for testing. The training and testing partition consists of approximately 16 and 2 hours of data for each language pair respectively.

4.2. Performance Measure

In the MSCS dataset, the ratio of primary and secondary language duration for each utterance has approximately $4 : 1$. Hence the use of accuracy, EER, and FER will provide biased performance toward the primary language. Similarly, SD literature suggests using JER instead of DER, if there exists a duration imbalance between the classes in the test utterances [20]. Therefore, JER is a better performance measure for evaluating the LD system performance. For comparison purposes, this study uses accuracy, EER, and DER along with JER to evaluate the performance of the LD systems.

4.3. Evaluation Setup

The pre-trained model available at [16] is used here. The pre-trained model is then fine-tuned using the training set of each language pair for 900 epochs. The pre-trained and fine-tuned models are detached from the gradient update and used as a feature extractor to train the W2V-ES and W2V-EA frameworks. Using the features extracted from the pre-trained network both W2V-ES and W2V-EA are trained. Similarly using the features extracted from the fine-tuned network both W2V-ES and W2V-EA are trained. All four models are trained for 60 epochs with 0.001 as the learning rate, 0.1 as dropout, and 32 as batch size. After training all four models are considered for testing and their results are compared. The codes of the implemented architectures are publicly available on GitHub.[1]. For comparison purposes, the x-vector-based E2E is replicated using the recipe and hyperparameter mentioned in [5]. Similarly, as proposed in [14], the standalone W2V fine-tuned architecture is also considered for testing and comparison.

4.4. Results and Discussion

The obtained results are presented in Table 2. Using the x-vector-based E2E (X-E2E) the average performance in terms of IDR, EER, DER, and JER is 81.3%, 6.4%, 22.2, and 60.4, respectively. Though the performance in terms of IDA is above 80%, the high difference between the DER and JER suggests the performance is biased towards primary language. Using W2V-F, as reported in [14], the performance is 82%, 5.3%, 24.3, and 36.8, respectively. These results show a relative improvement of 39% in terms of JER. It also shows that the difference between the DER and JER reduced as compared to the performance of the X-E2E system. This shows that the W2V can resolve the primary language bias issue to some extent.

The obtained results using pre-trained W2V features with W2V-ES in terms of IDA, EER, DER, and JER are 81.5%, 6.1%, 23.2, and 41.8, respectively. Using W2V-EA the performance is 81.4%, 6.1%, 22.8, and 41.4, respectively. It is observed that the performance of the E2E framework with pre-trained W2V features is inferior to the stand-alone W2V fine-tuned framework. However provides a relative improvement of 30.7% using SP and 31.4% using AP, over the performance of the X-E2E framework in terms of JER.

The performance obtained using the W2V fine-tuned feature using the W2V-ES framework is 89.1%, 3.5%, 13.2, and 22.9 in terms of IDA, EER, DER, and JER, respectively. Similarly using the W2V-EA framework, we have obtained 89.3%,

[1]https://github.com/jagabandhumishra/W2V-E2E-Language-Diarization

Table 2: *Performance comparison between W2V-E2E frameworks and baselines, P: pre-trained, F: Fine-tuned, RI: Relative improvement in terms of JER, Avg: language pair-wise averaged performance.*

Model	LP	IDA	EER	DER	JER	RI
X-E2E [5]	GUE	80.9	6.3	22.9	60.6	
	TAE	81.4	6.9	22.8	60.5	
	TEE	81.7	6.0	21.1	60.1	
	Avg	**81.3**	**6.4**	**22.2**	**60.4**	
W2V-F [14]	GUE	82.2	5.3	23.7	35.4	**39.0**
	TAE	80.9	5.6	25.0	37.2	
	TEE	82.9	5.1	24.2	37.8	
	Avg	**82**	**5.3**	**24.3**	**36.8**	
P-W2V-ES	GUE	83.1	5.6	22.3	40.5	**30.7**
	TAE	79.0	6.9	25.8	45.0	
	TEE	82.3	5.8	21.7	39.9	
	Avg	**81.5**	**6.1**	**23.2**	**41.8**	
P-W2V-EA	GUE	82.7	5.7	21.0	38.4	**31.4**
	TAE	80.8	6.3	23.7	43.5	
	TEE	80.6	6.4	23.8	42.3	
	Avg	**81.4**	**6.1**	**22.8**	**41.4**	
F-W2V-ES	GUE	89.2	3.5	12.8	23.0	**62.0**
	TAE	88.6	3.7	14.0	24.2	
	TEE	89.5	3.4	12.9	21.5	
	Avg	**89.1**	**3.5**	**13.2**	**22.9**	
F-W2V-EA	GUE	89.4	3.5	12.8	22.8	**62.6**
	TAE	88.7	3.7	13.9	23.8	
	TEE	89.9	3.3	11.3	20.9	
	Avg	**89.3**	**3.5**	**12.7**	**22.5**	
X-F-W2V-ES	GUE	90.0	3.3	12.1	22.8	**63.0**
	TAE	89.4	3.5	13.0	23.8	
	TEE	90.4	3.1	10.7	20.5	
	Avg	**89.9**	**3.3**	**11.9**	**22.3**	
X-F-W2V-EA	GUE	90.0	3.3	11.3	22.4	**63.9**
	TAE	89.8	3.5	11.8	23.2	
	TEE	90.5	3.1	10.7	20.0	
	Avg	**90.1**	**3.3**	**11.2**	**21.8**	

3.5%, 12.7, and 22.5, respectively. From the performance, it is observed that the system's performance further improves by considering fine-tuned W2V features. This improved architecture provides a relative improvement of 62% using SP and 62.6% using AP, over the X-E2E framework in terms of JER. Similarly, considering the identical framework as X-E2E, and only replacing the input of MFCC features with finetuned wav2vec features (X-F-W2V-ES) provides the performance of 89.9%, 3.3%, 11.9, and 22.3, respectively. The system provides a relative improvement of 63% over the baseline in terms of JER. Further, by modifying the aggregation strategy to AP (X-F-W2V-EA), the performance further improved and provide a relative improvement of 63.9% over the baseline in terms of JER.

The experimental observation is that the system using fine-tuned W2V features along with AP provides the best performance. It is also observed that the use of AP provides better performance than the SP. Further, to observe the language discrimination ability of the proposed models over the baseline X-E2E system, the embedding vectors are extracted just before the self-attention block and using t-SNE, projected to two dimensions and depicted in Figure 3. From the plot, it is observed that for all the plots the vectors belonging to silence mostly form a separate cluster, whereas the overlap between the primary and secondary language cluster reduces by moving from X-E2E to P-W2V-ES and is further reduced by moving to F-W2V-ES and F-W2V-EA framework.

The performance bias can be observed using the confusion matrix of the F-W2V-EA system along with baseline X-E2E and standalone W2V-F system as tabulated in Table 3. From the table, it can be observed that the F-W2V-EA system has the secondary to secondary language identification rate of 79.8%, followed by 63.9% and 0% using W2V-F and X-E2E system. Similarly, the secondary to primary language identification rate was reduced to 7.7% followed by 31.3% and 65.2% using W2V-F and X-E2E system. This study can be concludes that the use of fine-tuned W2V features using the E2E framework are able to resolve the primary language bias issue and improve the overall LD performance to 21.8 in terms of JER.

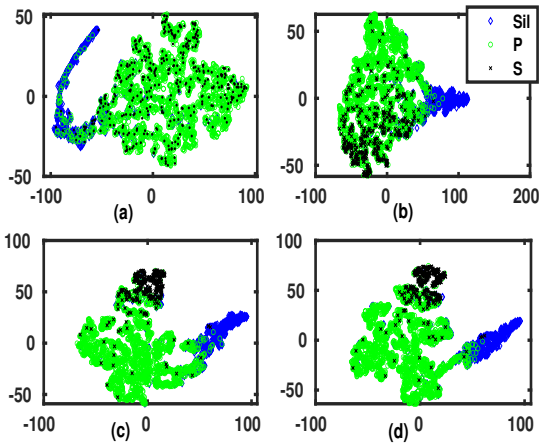

Figure 3: *The t-SNE distribution obtained from : (a) X-E2E, (b) P-W2V-ES, (c) F-W2V-ES, and (d) F-W2V-EA architectures, respectively.*

Table 3: *Comparison using confusion matrix, P: primary, S: secondary, and Sil: silence.*

Model		P	S	Sil
X-E2E	P	90.6	0	9.3
	S	65.2	**0**	34.7
	Sil	16.8	0	83.1
W2V-F	P	90.9	3.8	5.1
	S	31.3	**63.9**	4.6
	Sil	23	4.6	72.3
F-W2V-EA	P	85.5	4.3	10.1
	S	7.7	**79.8**	12.3
	Sil	5.8	3.8	90.3

5. Conclusion and Future work

This study reported the primary language bias of the existing systems. Inspired by the learning strategy of W2V, this work proposed an end-to-end framework using fine-tuned W2V architecture as a feature extractor. The proposed approach can resolve the primary language bias and provides a relative improvement of 63.9% over the baseline in terms of JER. The study also shows that while dealing with turn imbalance data, JER is a better measure to calibrate the LD system performance compared to DER, IDR, and EER. Further, it is also observed that irrespective of using pre-trained or fine-tuned features, the attention pooling-based temporal aggregation provides better performance over statistical pooling. In the future, the framework will be further explored to improve the performance of the LD system.

6. ACKNOWLEDGMENT

The authors like to acknowledge "Anatganak", IIT Dharwad, for enabling us to perform experiments, and the Ministry of Electronics and Information Technology (MeitY), Govt. of India, for supporting us through different projects.

7. References

[1] S. Sitaram, K. R. Chandu, S. K. Rallabandi, and A. W. Black, "A survey of code switching speech and language processing," *arXiv:1904.00784 [cs.CL]*, 2019.

[2] J. Mishra and S. R. M. Prasanna, "Challenges in spoken language diarization in code-switched scenario," in *2023 National Conference on Communications (NCC)*. IEEE, 2023, pp. 1–6.

[3] J. Mishra, A. Agarwal, and S. R. M. Prasanna, "Spoken language diarization using an attention based neural network," in *2021 National Conference on Communications (NCC)*. IEEE, 2021, pp. 1–6.

[4] V. Spoorthy, V. Thenkanidiyoor, and D. A. Dinesh, "SVM Based Language Diarization for Code-Switched Bilingual Indian Speech Using Bottleneck Features," in *Proc. The 6th Intl. Workshop on Spoken Language Technologies for Under-Resourced Languages*, 2018, pp. 132–136. [Online]. Available: http://dx.doi.org/10.21437/SLTU.2018-28

[5] H. Liu, L. P. G. Perera, X. Zhang, J. Dauwels, A. W. Khong, S. Khudanpur, and S. J. Styles, "End-to-end language diarization for bilingual code-switching speech," in *22nd Annual Conference of the International Speech Communication Association, INTERSPEECH 2021*, vol. 2. International Speech Communication Association, 2021.

[6] D. C. Lyu, E. S. Chng, and H. Li, "Language diarization for code-switch conversational speech," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 7314–7318.

[7] ——, "Language diarization for conversational code-switch speech with pronunciation dictionary adaptation," in *Signal and Information Processing (ChinaSIP), 2013 IEEE China Summit and International Conference on*. IEEE, 2013, pp. 147–150.

[8] G. Frost, E. Morris, J. Jansen van Vüren, and T. Niesler, "Fine-tuned self-supervised speech representations for language diarization in multilingual code-switched speech," in *Artificial Intelligence Research: Third Southern African Conference, SACAIR 2022, Stellenbosch, South Africa, December 5–9, 2022, Proceedings*. Springer, 2022, pp. 246–259.

[9] Y. Fujita, N. Kanda, S. Horiguchi, Y. Xue, K. Nagamatsu, and S. Watanabe, "End-to-end neural speaker diarization with self-attention," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2019, pp. 296–303.

[10] T. J. Park, N. Kanda, D. Dimitriadis, K. J. Han, S. Watanabe, and S. Narayanan, "A review of speaker diarization: Recent advances with deep learning," *Computer Speech & Language*, vol. 72, p. 101317, 2022.

[11] S. Shah, S. Sitaram, and R. Mehta, "First workshop on speech processing for code-switching in multilingual communities: Shared task on code-switched spoken language identification," *WSTCSMC 2020*, p. 24, 2020.

[12] D. Krishna and A. Patil, "Utterance-level code-switching identification using transformer network," *WSTCSMC 2020*, p. 53, 2020.

[13] P. Rangan, S. Teki, and H. Misra, "Exploiting spectral augmentation for code-switched spoken language identification," *arXiv preprint arXiv:2010.07130*, 2020.

[14] J. Mishra and S. R. M. Prasanna, "Importance of supra-segmental information and self-supervised framework for spoken language diarization task," in *International Conference on Speech and Computer*. Springer, 2022, pp. 494–507.

[15] J. Mishra, J. Gandra, V. Patil, and S. R. M. Prasanna, "Issues in sub-utterance level language identification in a code switched bilingual scenario," in *2022 IEEE International Conference on Signal Processing and Communications (SPCOM)*. IEEE, 2022, pp. 1–5.

[16] A. Gupta, H. S. Chadha, P. Shah, N. Chimmwal, A. Dhuriya, R. Gaur, and V. Raghavan, "Clsril-23: cross lingual speech representations for indic languages," *arXiv preprint arXiv:2107.07402*, 2021.

[17] 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.

[18] Z. Fan, M. Li, S. Zhou, and B. Xu, "Exploring wav2vec 2.0 on speaker verification and language identification," *arXiv preprint arXiv:2012.06185*, 2020.

[19] Z. Bai and X.-L. Zhang, "Speaker recognition based on deep learning: An overview," *Neural Networks*, vol. 140, pp. 65–99, 2021.

[20] N. Ryant, K. Church, C. Cieri, A. Cristia, J. Du, S. Ganapathy, and M. Liberman, "First dihard challenge evaluation plan," *2018, tech. Rep.*, 2018.