End-to-end language diarization for bilingual code-switching speech

Liu, Hexin; Perera, Leibny Paola Garcia; Zhang, Xinyi; Dauwels, Justin; Khong, Andy W.H.; Khudanpur, Sanjeev; Styles, Suzy J.

DOI
10.21437/Interspeech.2021-82

Publication date
2021

Document Version
Final published version

Published in
22nd Annual Conference of the International Speech Communication Association, INTERSPEECH 2021

INTERSPEECH 2021
30 August – 3 September, 2021, Brno, Czechia

End-to-End Language Diarization for Bilingual Code-Switching Speech

Hexin Liu[1], *Leibny Paola Garcia Perera*[2], *Xinyi Zhang*[1], *Justin Dauwels*[3], *Andy W. H. Khong*[1], *Sanjeev Khudanpur*[2], *Suzy J. Styles*[4]

[1]School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore
[2]CLSP and HLT-COE, Johns Hopkins University, USA
[3]Department of Microelectronics, Delft University of Technology, Netherlands
[4]Psychology, School of Social Sciences, Nanyang Technological University, Singapore

HEXIN002@e.ntu.edu.sg, lgarci27@jhu.edu

Abstract

We propose two end-to-end neural configurations for language diarization on bilingual code-switching speech. The first, a BLSTM-E2E architecture, includes a set of stacked bidirectional LSTMs to compute embeddings and incorporates the deep clustering loss to enforce grouping of languages belonging to the same class. The second, an XSA-E2E architecture, is based on an x-vector model followed by a self-attention encoder. The former encodes frame-level features into segment-level embeddings while the latter considers all those embeddings to generate a sequence of segment-level language labels. We evaluated the proposed methods on the dataset obtained from the shared task B in WSTCSMC 2020 and our handcrafted simulated data from the SEAME dataset. Experimental results show that our proposed XSA-E2E architecture achieved a relative improvement of 12.1% in equal error rate and a 7.4% relative improvement on accuracy compared with the baseline algorithm in the WSTCSMC 2020 dataset. Our proposed XSA-E2E architecture achieved an accuracy of 89.84% with a baseline of 85.60% on the simulated data derived from the SEAME dataset.
Index Terms: language diarization, language identification, code-switching, end-to-end neural diarization, self-attention

1. Introduction

The term diarization was originally used to describe the task of determining audio segments associated with the same speaker in a multi-speaker recording. It was subsequently extended to cover language diarization (LD), a special case of language identification (LID), where the task is to identify the languages of each utterance in a natural multilingual recording. This work addresses LD in the context of code-switching speech (cf e.g. [1]) where the same speaker may use more than one language, often within the same utterance.

Traditional speaker diarization methods, which aim to segment the speech signal and group together segments belonging to the same speaker, are traditionally based on a speaker encoding front-end, such as x-vectors [2], and a clustering back-end model [3–8]. These are inherently unsupervised methods, in the sense that they neither require nor leverage examples of segmented-and-labeled multi-speaker recordings [9]. End-to-end (E2E) speaker diarization methods, by contrast, require (and learn from) labeled multi-speaker audio to jointly train an integrated neural module for speaker encoding and clustering [9–12].

Traditional LID methods, which assume that each audio segment presented to the system contains speech in one language that must be identified [13], are similarly based on a two-stage process [14–16], where a language embedding such as an x-vector is first extracted from the speech using a front-end encoder, and a separately trained classifier is then employed as the back-end to identify the language using the embedding. Again, by contrast, recently proposed E2E neural LID methods [17–20] integrate this two-stage process into a single neural module. They work in a similar manner as E2E speaker diarization —initial layers of a deep neural network (DNN) generate an embedding for the input speech, and subsequent DNN layers perform language classification.

Note that language diarization cannot be done using LID methods due to the one-language-per-audio-sample assumption. Instead, inspired by E2E speaker diarization [10, 11] and the x-vector language encoder [15], we propose two methods for E2E neural language diarization. The first approach uses a bidirectional long short-term memory neural network [21] to build an end-to-end LD model (BLSTM-E2E), as originally proposed for speaker diarization in [10]. This model is jointly trained for language diarization by minimizing the cross-entropy (CE) loss and a deep-clustering (DC) loss [22]. The second employs an x-vector network followed by a self-attention transformer encoder [23] to build an end-to-end LD model (XSA-E2E). The x-vector layers encode frame-level features into a sequence of segment-level embeddings, from which the self-attention transformer generates a sequence of segment-level language labels. For completeness, we also compare the performance of these two models with intermediate model that uses only a self-attention module on top of the BLSTM (SA-E2E).

These approaches have three desirable properties. Firstly, language diarization is a multi-label classification problem, for which these models appropriately generate a sequence of segment-level labels for each test recording, where a segment consists of several adjacent frames, allowing the model to identify different languages within an utterance, detect the language change point, and tag the silences in a unified manner. Secondly, as opposed to diarization approaches that require pre-processing for speech-activity detection (SAD), these proposed models perform SAD implicitly by defining silence as an output label. Finally, hierarchical processing is employed in the proposed XSA-E2E model with the multi-objective training. This hierarchical processing captures local language information in each segment before establishing global dependency between input and output, which benefits language diarization. We evaluated our proposed approaches on the code-switching data set from the Shared Task B of the First Workshop on Speech Technologies for Code-Switching in Multilingual Communities (WSTCSMC 2020) [24], and on simulated data derived from the SEAME data set [25].

http://dx.doi.org/10.21437/Interspeech.2021-82

2. Proposed Methods

2.1. BLSTM-based end-to-end model (BLSTM-E2E)

We apply the BLSTM-E2E model that was originally proposed in [10] for speaker diarization to language diarization. Consider a sequence of segments, where we define $\mathbf{X} = (\mathbf{x}_t \in \mathbb{R}^B | t = 1, ..., T)$ as features extracted from those segments with T being the number of segments and B the dimension of the segment-level feature vector. The ground-truth language label sequence is defined as $\mathbf{Y} = (\mathbf{y}_t | t = 1, ..., T)$, where the class label $\mathbf{y}_t \in \{0, 1, ..., C\}$ given that 0 is the label corresponding to a silent segment when SAD task is included and C is the total number of languages. In this architecture, the first N BLSTM layers generate language representations (embeddings) for each segment and the next M BLSTM layers estimate the label sequence for these embeddings.

To apply the DC loss [22], a D-dimensional embedding vector $\mathbf{e}_t$ is transformed from the hidden activations $\mathbf{h}_t^N$ of the N-th BLSTM layer. We replaced the permutation-free loss used in [10] with the cross-entropy loss and the multi-objective loss can be then computed via

$$L^{\mathrm{CE}} = \mathrm{CE}\left(\mathbf{Y}, \hat{\mathbf{Y}}\right), \quad (1)$$

$$L^{\mathrm{BLSTM}} = \alpha L^{\mathrm{CE}} + (1 - \alpha) L^{\mathrm{DC}}. \quad (2)$$

Here, L denotes the loss, $\mathrm{CE}(\cdot, \cdot)$ denotes the cross-entropy loss function between the ground-truth label sequence $\mathbf{Y}$, the output $\hat{\mathbf{Y}} = [\hat{\mathbf{y}}_1 \cdots \hat{\mathbf{y}}_t]^\intercal$ of the BLSTM-E2E model and α is a scaling factor to facilitate multi-objective training.

2.2. Self-attention-based end-to-end model (SA-E2E)

We first explore the SA-E2E model as a preliminary method to understand the XSA-E2E model. This SA-E2E model employs the encoder module of the transformer in [23]. The SA-E2E model comprises the positional encoding, encoder blocks, and a linear layer with sigmoid activation function. The input $\mathbf{X} = (\mathbf{x}_t \in \mathbb{R}^B | t = 1, ..., T)$ of SA-E2E is the same as that of the BLSTM-E2E model and the output $\hat{\mathbf{Y}} = [\hat{\mathbf{y}}_1 \cdots \hat{\mathbf{y}}_t]^\intercal$ is computed via

$$\hat{\mathbf{Y}} = \mathrm{Encoder}(\mathbf{X}), \quad (3)$$

where $\mathrm{Encoder}(\cdot)$ denotes the SA-E2E model. The architecture of the SA-E2E model is presented as the self-attention encoder module in Fig. 1.

2.3. XSA end-to-end model (XSA-E2E)

The x-vector has shown to achieve high performance on short-utterance LID [15] by capturing local language information. The transformer, on the other hand, was proposed to draw global dependencies between input and output [23]. Considering the benefits of both techniques, we employ the x-vectors to capture the local information of each segment and the transformers that take the temporal dynamics of the signal into account.

For a speech signal partitioned in segments, each $\mathbf{x}_t$ in the input sequence $\mathbf{X} = (\mathbf{x}_t \in \mathbb{R}^{K \times F} | t = 1, ..., T)$ of the XSA-E2E model is a matrix $[\mathbf{f}_1, ..., \mathbf{f}_K]^\intercal$, where $\mathbf{f}_K$ is an F-dimensional frame-level feature vector of the K-th frame in segment t. The proposed XSA-E2E model operates in a hierarchical manner as shown in Fig. 1. It first processes frame-level into segment-level features before estimating the posterior for each segment.

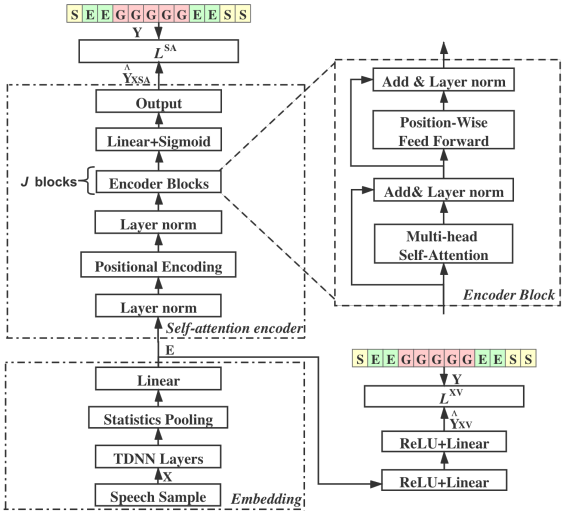

Figure 1: *X-vector-Self-Attention end-to-end (XSA-E2E) language diarization model with multi-objective training.*

Frame-level features of each $\mathbf{x}_t$ are encoded into a segment-level embedding through the x-vector embedding module. The use of x-vector embedding allows the algorithm to capture the language identity of each segment. We first define the segment-level embeddings $\mathbf{E} = (\mathbf{e}_t \in \mathbb{R}^D | t = 1, ..., T)$, which are computed via

$$\mathbf{E} = \mathrm{Embedding}(\mathbf{X}), \quad (4)$$

where $\mathrm{Embedding}(\cdot)$ denotes the embedding module in Fig. 1. The outputs of the x-vector model and the XSA-E2E model are then computed, respectively as

$$\hat{\mathbf{Y}}_{\mathrm{XSA}} = \mathrm{Encoder}(\mathbf{E}), \quad (5)$$

$$\hat{\mathbf{Y}}_{\mathrm{XV}} = \mathrm{Xvector}(\mathbf{X}), \quad (6)$$

where $\mathrm{Encoder}(\cdot)$ is the encoder module in Fig. 2 and $\mathrm{Xvector}(\cdot)$ is the x-vector model employed in our proposed XSA-E2E model. XV and XSA denote the x-vector model and the XSA-E2E model in Fig. 1, respectively.

Similar to BLSTM-E2E, the proposed XSA-E2E employs a multi-objective training paradigm. Inspired by the cross-entropy loss function of the TDNN-based x-vector model in [15], we adopted the cross-entropy loss for the XSA-E2E model; consequently, the multi-objective loss function is computed as

$$L^{\mathrm{XV}} = \mathrm{CE}\left(\mathbf{Y}, \hat{\mathbf{Y}}_{\mathrm{XV}}\right), \quad (7)$$

$$L^{\mathrm{SA}} = \mathrm{CE}\left(\mathbf{Y}, \hat{\mathbf{Y}}_{\mathrm{XSA}}\right), \quad (8)$$

$$L^{\mathrm{XSA}} = \beta L^{\mathrm{XV}} + (1 - \beta) L^{\mathrm{SA}}, \quad (9)$$

where $\mathbf{Y} = (\mathbf{y}_t | t = 1, ..., T)$ is the ground-truth language label sequence, β is the parameter for multi-objective training.

3. Experiments

3.1. Dataset

We conducted experiments on two datasets. The first dataset is obtained from the shared task B in WSTCSMC 2020 comprising three code-switching language pairs: Gujarati-English (gu-en), Tamil-English (ta-en) and Telugu-English (te-en). The proposed model for each language was trained on its corresponding

Table 1: *Duration (hours) of each language in the shared task B in WSTCSMC 2020*

Language pair	Training			Development		
	gu/ta/te	en	silence	gu/ta/te	en	silence
gu-en	10.6	2.1	3.0	1.3	0.3	0.4
ta-en	10.9	1.8	3.1	1.4	0.2	0.4
te-en	10.9	1.8	3.0	1.4	0.2	0.4

Table 2: *Utterances description of the simulated data using SEAME dataset*

Simulated data	Num.of classes in utterances		
	one class	two classes	three classes
no silence	7556	4936	-
with silence	3271	8521	3921

training set and evaluated on the development set. Detailed information of this dataset [24] is shown in Table 1. In addition, to verify the robustness and the extension capability to multiple classes, our models were trained on all three language pairs to achieve a five-class language diarization task consisting of English, Gujarati, Tamil, Telugu, and silence. Test results are then reported on the development sets of these three code-switching pairs.

The algorithms were also validated on the SEAME dataset [25]. This SEAME dataset comprises 290 recordings consisting of 110,145 utterances out of which 24,438 are Mandarin, 28,655 are English, and 57,052 are code-switching utterances. Since there are no time-stamps corresponding to the language change for these Mandarin-English code-switching utterances, code-switching data was simulated by concatenating no more than five monolingual utterances from the same recording in chronological order. The maximum length of the final utterance was set to 50 s. A total of 18.7-hour of English, 18.1-hour of Mandarin and 5.7-hour of silence were used to build the simulated data. For the version without silence, we used 12,492 speech samples with an average duration of 10.6 s. For the simulated data with silence, we occasionally interleaved silence segments and we used 15,713 simulated speech samples with an average duration of 9.7 s. In contrast to the WSTCSMC 2020 dataset, in which each utterance comprises three classes, the simulated data utterances can contain one, two or three classes. The related information is shown in Table 2. In addition, since there is no partition on development or test, we conducted a ten-fold cross validation.

3.2. Experiment setup

3.2.1. Feature extraction

Since the ground-truth language labels of the data in WSTCSMC 2020 are assigned to 200 ms segments, the input speech sample is first partitioned into segments of the same duration. We ignore the remaining of the speech samples that cannot be divided exactly into 200ms segments. For each segment, 23-dimensional log-Mel-filterbank features are extracted with a 25 ms window and a 10 ms shift as [10] for all systems.

The 19-frame log-Mel-filterbank features are then processed into segment-level features in two different forms. The first directly concatenates features of all 19 frames into a 437-dimensional vector before feeding the vector into our E2E models. The second employs a pruned x-vector model [15] as a feature extractor. A 256-dimensional embedding is extracted as the segment-level input of the encoding module.

Table 3: *Comparison of our approaches with DeepSpeech2 system [24, 27] and Vocapia-LIMSI system [28] on the dev set of the shared task B in WSTCSMC 2020 by employing EER (%) and Accuracy (%)*

Method	gu-en		ta-en		te-en		Average	
	EER	Acc.	EER	Acc.	EER	Acc.	EER	Acc.
DeepSpeech2 [24, 27]	6.7	76.7	6.5	77.6	6.7	76.5	6.6	76.9
Vocapia-LIMSI [28]	-	80.5	-	81.2	-	81.8	-	81.2
BLSTM-E2E	6.1	81.8	5.9	82.4	**5.7**	**82.8**	5.9	82.3
SA-E2E	6.4	80.9	6.1	81.6	6.0	81.9	6.2	81.5
XSA-E2E	**5.8**	**82.7**	**5.9**	**82.4**	5.8	82.6	**5.8**	**82.6**

3.2.2. Model configuration

The BLSTM-E2E language diarization model employs five BLSTM layers with 256 hidden units in each layer. The outputs of the second BLSTM layer are transformed into 256-dimensional vectors for the computation of DC loss and $\alpha = 0.5$ in (2). We used the Adam optimizer [26] with an initial learning rate of 10^{-3} and cosine annealing learning rate decay. The model was trained for 60 epochs with a batch size of 8.

For the XSA-E2E language diarization model, we applied an x-vector model followed by a self-attention encoder. The x-vector model is the same as that in [15] except that the third TDNN layer was removed to adapt to the length of segment and the dimension of the x-vector embedding $\mathbf{e}_t$ is given by $D = 256$. As shown in Fig. 1, the 256-dimensional x-vectors are fed into the self-attention encoder module. This module comprises four encoder blocks with four heads in each multi-head self-attention layer ($J = 4$) and a position-wise feed forward layer with 2048 hidden units in the inner-layer. We used the Adam optimizer with an initial learning rate of 10^{-4} with cosine annealing learning rate decay. The model was trained for 30 epochs with a batch size of 32.

The SA-E2E model which is equivalent to the self-attention encoder module of our XSA-E2E model is also implemented as baseline. The Adam optimizer was applied to train the SA-E2E model for 60 epochs in total with an initial learning rate of 10^{-4} which decays after 10 warm-up epochs. The source code for this research is made publicly available in GitHub. [1]

We evaluated our systems using accuracy and equal error rate (EER). To compare with the models in WSTCSMC 2020, we applied the method described in this workshop [24] to compute the accuracy and EER.

3.3. Results

3.3.1. Evaluation on WSTCSMC 2020 shared task B

The results for the shared task B in WSTCSMC 2020 are shown in Table 3. The baseline models include DeepSpeech2 [24, 27] and Vocapia-LIMSI system [28], where the latter is a fusion system of an unsupervised GMM-based i-vector model [29] and a phonotactic model. As shown in Table 3, both BLSTM-E2E and XSA-E2E models outperformed baseline algorithms on this dataset; the proposed BLSTM-E2E system achieved the best performance on Telugu-English code-switching data while the XSA-E2E system achieved the best performance on Gujarati-English and Tamil-English code-switching data.

To validate the performance of our proposed models under a more challenging condition, we pooled together code-switching data of all three language pairs as the training data. The results presented in Table 4 show that the XSA-E2E model achieves

[1] https://github.com/Lhx94As/E2E-langauge-diarization

Table 4: *Comparison of our approaches on 3-language-pair code-switching data in WSTCSMC 2020 by employing EER (%) of each language and Accuracy (%)*

Method	en	gu	ta	te	silence	Accuracy
BLSTM-E2E	6.27	3.94	3.55	3.52	**2.97**	80.15
SA-E2E	6.33	3.59	3.73	3.65	3.49	79.21
XSA-E2E	**5.99**	**2.98**	**3.21**	**3.05**	3.56	**81.20**

Table 5: *10-fold cross validation results of our approaches on simulated code-switching data using SEAME dataset by employing EER (%) and Accuracy (%). EER for the simulated data with silence is composed of EER for English (Eng), EER for Mandarin (Man), and EER for Silence (Sil)*

Method	Simulated data using SEAME					
	no silence		with silence			
	EER	Acc.	Eng	Man	Sil	Acc.
BLSTM-E2E	7.20	85.60	6.47	6.37	**0.93**	86.23
SA-E2E	7.15	85.71	6.47	6.25	1.67	85.60
XSA-E2E	**5.08**	**89.84**	**5.06**	**4.91**	2.38	**87.66**

the highest overall accuracy among the three proposed models and lowest EER on all classes except silence. It is worth noting that although the SA-E2E model suffers from the worst overall performance, it achieves similar performance to that of BLSTM-E2E on four language classes. It also achieves worse performance on silence segments than BLSTM-E2E. These results imply that the self-attention mechanism may not perform as well as the BLSTM model for data with silence. The performance of XSA-E2E is reduced further compared to SA-E2E by the silent segments. This is not surprising given that about 20% of the data from WSTCSMC 2020 is silence [24]. Moreover, English only accounts for 12% of this dataset, leading to high EER on English for all models in this experiment.

3.3.2. Evaluation on simulated code-switching data using SEAME dataset

We evaluated our approaches on two types of simulated data. The results presented in Table 5 show that our proposed XSA-E2E system achieves the best performance on both types of simulated data in this evaluation. In addition, both XSA-E2E and SA-E2E models which employ the self-attention mechanism perform worse on silence segments than the BLSTM-E2E model. After including silences in the simulated data, XSA-E2E and SA-E2E exhibit degradation of accuracy when compared with the BLSTM-E2E model. The XSA-E2E model suffers from the highest EER on the silence data. These observations are consistent with results presented in Table 3 and Table 4.

3.4. Analysis and visualization of self-attention heads

To investigate the operation of the self-attention mechanism for language diarization task, we analyze the attention weight matrices of two attention heads of the second encoder block in the XSA-E2E model, shown in Fig. 2. The attention weights in the left head lead to a linear transformation, while the right head horizontally exhibits different color depths for different classes. This implies that the self-attention mechanism in our proposed XSA-E2E model is able to capture the language identity for language diarization and speech activity detection.

In addition, Fig. 2 shows how data composition influences our proposed XSA-E2E model. The XSA-E2E model trained on the simulated version of the SEAME dataset shows clearer

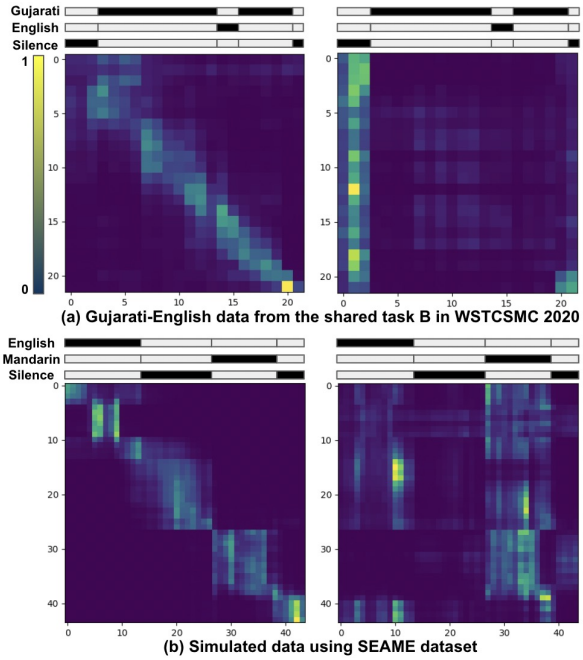

Figure 2: *Attention weights at the second encoder block of the XSA-E2E model trained on (a) Gujaratu-English code-switching data and (b) simulated data using SEAME dataset with silence.*

boundaries of different classes in the right head than trained on the Gurajarati-English data. This is due to the simulated data being more balanced than the the WSTCSMC 2020 dataset resulting in higher accuracy and lower EER.

4. Conclusions

We proposed the BLSTM-E2E and the XSA-E2E models for the language diarization task on bilingual code-switching speech. Our proposed XSA-E2E model improves the state-of-the-art performance on the development set of the shared task B in WSTCSMC 2020 and achieves the best performance on simulated data derived from the SEAME dataset. Compared to SA-E2E model, our proposed models that employ embedding-related loss for joint training achieve higher performance in most experiments. Compared to the SA-E2E model, the XSA-E2E also achieves higher performance with a hierarchical processing. These underpin the importance of the local information in each segment for the language diarization task. The results also highlight that both x-vector and self-attention mechanisms can perform higher on data with less silence. We also show how self-attention captures the language characteristics through the attention weights. Our model may be employed as a preprocessing module of a multilingual speech recognition system. In addition, the research into improving the performance of language diarization systems on silence frames can be interesting as future work.

5. Acknowledgements

This work was supported of the National Research Foundation, Singapore, under the Science of Learning programme (NRF2016-SOL002-011), and the Centre for Research and Development in Learning (CRADLE) at Nanyang Technological University, Singapore (JHU IO 90071537).

6. References

[1] D. Lyu, C. E. Siong, and H. Li, "Language diarization for code-switch conversational speech," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2013, pp. 7314–7318.

[2] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2018, pp. 5329–5333.

[3] S. H. Shum, N. Dehak, R. Dehak, and J. R. Glass, "Unsupervised methods for speaker diarization: An integrated and iterative approach," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 21, no. 10, pp. 2015–2028, 2013.

[4] G. Sell and D. Garcia-Romero, "Speaker diarization with plda i-vector scoring and unsupervised calibration," in *Proc. IEEE Spoken Language Technology Workshop*, 2014, pp. 413–417.

[5] D. Garcia-Romero, D. Snyder, G. Sell, D. Povey, and A. McCree, "Speaker diarization using deep neural network embeddings," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2017, pp. 4930–4934.

[6] Q. Wang, C. Downey, L. Wan, P. A. Mansfield, and I. L. Moreno, "Speaker diarization with lstm," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2018, pp. 5239–5243.

[7] H. Ning, M. Liu, H. Tang, and T. S. Huang, "A spectral clustering approach to speaker diarization," in *Ninth International Conference on Spoken Language Processing*, 2006.

[8] M. Diez, F. Landini, L. Burget, J. Rohdin, A. Silnova, K. Žmolíková, O. Novotný, K. Veselý, O. Glembek, O. Plchot, L. Mošner, and P. Matějka, "But system for dihard speech diarization challenge 2018," in *Proc. Interspeech*, 2018, pp. 2798–2802.

[9] A. Zhang, Q. Wang, Z. Zhu, J. Paisley, and C. Wang, "Fully supervised speaker diarization," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2019, pp. 6301–6305.

[10] Y. Fujita, N. Kanda, S. Horiguchi, K. Nagamatsu, and S. Watanabe, "End-to-End Neural Speaker Diarization with Permutation-free Objectives," in *Proc. Interspeech*, 2019, pp. 4300–4304.

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

[12] G. Bhattacharya, J. Monteiro, J. Alam, and P. Kenny, "Generative adversarial speaker embedding networks for domain robust end-to-end speaker verification," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2019, pp. 6226–6230.

[13] H. Li, B. Ma, and K. A. Lee, "Spoken language recognition: from fundamentals to practice," *Proceedings of the IEEE*, vol. 101, no. 5, pp. 1136–1159, 2013.

[14] N. Dehak, P. A. Torres-Carrasquillo, D. Reynolds, and R. Dehak, "Language recognition via i-vectors and dimensionality reduction," in *Proc. Twelfth Annual Conf. Int. Speech Comm. Assoc.*, 2011.

[15] D. Snyder, D. Garcia-Romero, A. McCree, G. Sell, D. Povey, and S. Khudanpur, "Spoken language recognition using x-vectors." in *Proc. Odyssey*, 2018, pp. 105–111.

[16] V. Mingote, D. Castan, M. McLaren, M. K. Nandwana, A. O. Giménez, E. Lleida, and A. Miguel, "Language recognition using triplet neural networks." in *Proc. Interspeech*, 2019, pp. 4025–4029.

[17] W. Cai, D. Cai, S. Huang, and M. Li, "Utterance-level end-to-end language identification using attention-based cnn-blstm," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2019, pp. 5991–5995.

[18] B. Padi, A. Mohan, and S. Ganapathy, "Attention based hybrid i-vector blstm model for language recognition." in *Proc. Interspeech*, 2019, pp. 1263–1267.

[19] X. Miao, I. McLoughlin, and Y. Yan, "A New Time-Frequency Attention Mechanism for TDNN and CNN-LSTM-TDNN, with Application to Language Identification," in *Proc. Interspeech*, 2019, pp. 4080–4084.

[20] L. Wan, P. Sridhar, Y. Yu, Q. Wang, and I. L. Moreno, "Tuplemax loss for language identification," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2019, pp. 5976–5980.

[21] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional lstm and other neural network architectures," *Neural networks*, vol. 18, no. 5-6, pp. 602–610, 2005.

[22] J. R. Hershey, Z. Chen, J. Le Roux, and S. Watanabe, "Deep clustering: Discriminative embeddings for segmentation and separation," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2016, pp. 31–35.

[23] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. NIPS*, 2017, pp. 5998–6008.

[24] 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," *WSTC-SMC 2020*, p. 24, 2020.

[25] D. C. Lyu, T. P. Tan, E. S. Chng, and H. Li, "SEAME: a Mandarin-English code-switching speech corpus in south-east asia," in *Proc. Interspeech*, 2010, pp. 1986–1989.

[26] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. ICLR*, 2015.

[27] D. Amodei, S. Ananthanarayanan, R. Anubhai, J. Bai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, Q. Cheng, G. Chen *et al.*, "Deep speech 2: End-to-end speech recognition in english and mandarin," in *Proc. ICML*. PMLR, 2016, pp. 173–182.

[28] B. Claude, L. Viet-Bac, and G. Jean-Luc, "Vocapia-limsi system for 2020 shared task on code-switched spoken language identification," in *The First Workshop on Speech Technologies for CodeSwitching in Multilingual Communities*, 2020.

[29] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 19, no. 4, pp. 788–798, 2010.