Multi-Target Filter and Detector for Speaker Diarization

Chin-Yi Cheng\textsuperscript{1}, Hung-Shin Lee\textsuperscript{1}, Yu Tsao\textsuperscript{2}, and Hsin-Min Wang\textsuperscript{1}

\textsuperscript{1}Institute of Information Science, Academia Sinica
\textsuperscript{2}Research Center for Information Technology Innovation, Academia Sinica

\{james31423a, hungshinlee\}@gmail.com

Abstract

A good representation of a target speaker usually helps to extract important information about the speaker and detect the corresponding temporal regions in a multi-speaker conversation. In this paper, we propose a neural architecture that simultaneously extracts speaker embeddings consistent with the speaker diarization objective and detects the presence of each speaker frame by frame, regardless of the number of speakers in the conversation. To this end, a residual network (ResNet) and a dual-path recurrent neural network (DPRNN) are integrated into a unified structure. When tested on the 2-speaker CALLHOME corpus, our proposed model outperforms most methods published so far. Evaluated in a more challenging case of concurrent speakers ranging from two to seven, our system also achieves relative diarization error rate reductions of 26.35\% and 6.4\% over two typical baselines, namely the traditional x-vector clustering system and the attention-based system.

Index Terms: speaker diarization, speaker embeddings

1. Introduction

Speaker diarization is the process of determining when individual speakers are active in a recording. The aim is to generate a diary of the presence of each speaker at each point in time. It also plays an integral role in the pre-processing stage of automatic speech recognition (ASR) and understanding. This technique has been widely used for speech processing in various scenarios, such as conference conversations, broadcast news, debates, and cocktail parties [1]. However, the resilience of speaker diarization remains weak due to the challenges posed by variations in recording channel, environment, reverberation, ambient noise, and the number of speakers [2].

For nearly a decade, researchers have tackled diarization problems using probabilistic models [3] or neural networks [4]. Many methods involve two steps, segmentation and clustering. In the segmentation step, a 1.5-second sliding window (with 50\% overlap) is typically used to divide a session into a sequence of short segments. Then, a speaker model is used to extract the speaker representation (e.g., the x-vector [5, 6, 7, 8], i-vector [9, 10], or d-vector [11, 12]) of each segment. During the clustering process, segments with highly homogeneous characteristics form a group. Different clustering techniques have been used according to various similarity measures, such as probabilistic linear discriminant analysis (PLDA) and cosine similarity [13, 14, 15, 10]. For example, agglomerative hierarchical clustering (AHC) and spectral clustering (SC) were used in [6, 10] and [16, 17], respectively. The unbounded interleaved-state recurrent neural network (UIS-RNN), stemming from the Gaussian mixture model (GMM) [18, 19] and hidden Markov model (HMM) [7], was used in [12]. In addition, some post-processing methods, such as Variational Bayes (VB) [20] and the LSTM-based method [21], have been applied to refine the initial diarization results.

Recently, some research [22, 23, 24, 25] has focused on end-to-end speaker diarization. Fujita et al. [22] reformulated the diarization task as a multi-label classification problem and utilized the permutation-invariant training (PIT) [26] technique in model training. Moreover, in [27], reliable speaker embeddings were derived by a selector to help a voice activity detector diarize a session. Self-attention [28, 29] and frame selection [30] were also used in end-to-end speaker diarization.

The traditional two-step method cannot handle overlapping speech in a session. Target speaker voice activity detection (TS-VAD) [31], originally designed to handle multi-speaker ASR, achieved good performance on overlapping speech processing. It relies on a x-vector/SC procedure [5, 8] to provide first-stage timestamps of each “target” (active) speaker’s speech, which are used to extract the first-stage i-vector for each target speaker from frames where only a single speaker is active. Finally, TS-VAD uses the i-vectors of all speakers and MFCCs to generate diarization results. Unfortunately, TS-VAD can only be applied to sessions with a fixed number of speakers, because its neural structure contains tensor concatenation of speaker embeddings. Inspired by the dual-path recurrent neural network (DPRNN) [32, 33], an outstanding model in speech separation, we propose a unified structure called Multi-Target Filter and Detector (MTFAD) that can handle conversations with various speaker numbers with a single model. Furthermore, finding that the quality of speaker representations has an impact on the diarization performance, we extend TS-VAD by building with a neural filter that can directly extract speaker representations suitable for the diarization task.

The main contribution of this paper is threefold. First, we greatly extend the practical scope of TS-VAD while inheriting its excellent performance in the speaker diarization task. In this sense, MTFAD has an advantage over TS-VAD because MTFAD does not set a limit on the number of speakers in a session. In addition to expanding practical use, since training data of different speaker numbers can be used together to train a single model, the model is more powerful than multiple separate models, each trained with training data of a specific number of speakers. Second, we design a filter that can be jointly trained with the diarization model to extract speaker embeddings. With this filter, we not only get better diarization performance, but also avoid pre-training the i-vector extractor. Third, unlike some previous studies, such as [6, 12], which deal with only non-overlapping speech, our model also performs well on data containing overlapping speech regions.

2. Multi-Target Filter and Detector

2.1. TS-VAD

TS-VAD [31] adopts a two-step diarization approach. The first step relies on traditional x-vector/SC diarization methods to
generate first-stage prediction timestamps for each speaker, and then these timestamps can be used to obtain the i-vector for each speaker. The i-vector is extracted by a pre-trained speaker model (i-vector extractor) from the frame-level MFCCs corresponding only to the target speaker. The second step requires two inputs, i.e., the frame-level MFCCs and the i-vectors of all target speakers. Similar to the structure shown in Fig. 1, the MFCCs are first passed through a four-layer convolutional neural network (CNN) and then respectively concatenated to each speaker’s i-vector to form the input of a BiLSTM. The output of the BiLSTM for each speaker is concatenated and fed into another BiLSTM for final classification. The spec of the second BiLSTM depends on the number of speakers in a session and the number of target speakers, respectively. The BiLSTM module works in a cascaded way. Instead of forming a 2-dimensional shape of $D \times T \times N$, we can replace the second BiLSTM in the cascaded BiLSTM module with multiple BiLSTM modules. By recursively processing data through the intra-block and inter-block, DRPRNN provides good speech separation results.

Inspired by DRPRNN, we design a detector that can handle different numbers of speakers. In speech separation, the length of each session is different, so the number of blocks ($N$) in the DRPRNN framework is a variable. As shown in Fig. 1, by treating SDEs as intra-block $(D \times T)$ and the number of speakers as the variable $N$, we can replace the second BiLSTM in TS-VAD with DRPRNN to obtain the detector in MTFAD. The SDEs are generated in the same way as in the TS-VAD framework. Instead of forming a 2-dimensional shape of $T \times ND$ by concatenating along dimension $D$, we combine $N$ matrices of $D \times T$ into a 3-dimensional multi-SDE tensor of $D \times T \times N$, where $D$, $T$, and $N$ denote the number of hidden channels, the number of frames in a session, and the number of target speakers, respectively. The $D \times T \times N$ multi-SDE tensor is the input to the dual-path BiLSTM module.

Each SDE with size $D \times T \times N$ is the number of channels, and $N$ is the number of sliding windows. The 3-dimensional tensor is converted by the “overlap-add” operation into consecutive outputs. The BiLSTM module goes through two dimensions: the size of the chunks (intra-block RNN) and the number of chunks (inter-block RNN). As a result, the RNNs can see information around and far away from the current time frame.

The dual-path BiLSTM module consists of an intra-block and an inter-block in order. Both blocks contain BiLSTM, Linear, and LayerNorm in a row. The output of each block is a residual addition of the input and output of LayerNorm. The only difference between these two blocks is their input. $X_{\text{input}}$ is segmented into $N$ intra-chunks $(D \times T)$. All intra-chunks, i.e., $X_{\text{input}}$, are input to the intra-block. The output of the intra-block, $X_{\text{intra}}$, is resegmented into $T$ inter-chunks $(D \times N)$, which are input to the inter-block. The output of the inter-block, $X_{\text{inter}}$, is used as the input of the next dual-path BiLSTM module. By recursively processing data through the intra-block and inter-block, DRPRNN provides good speech separation results.

Figure 1: The structure of MTFAD, where $D$, $N$, $T$, and $N$ denote the concatenation operator, number of speakers, embedding size, and frame numbers, respectively. $\otimes$ is an element-wise product operator of MFCCs and each mask (labels of speaker occurrences) generated from the first stage.

Figure 2: A DRPRNN module contains an intra-block and an inter-block. Operator $+$ denotes the addition of two tensors. $X_{\text{intra}}$, $X_{\text{intra}}$, and $X_{\text{inter}}$ have the same shape. In case of cascaded DRPRNN modules, the preceding $X_{\text{intra}}$ is the direct $X_{\text{input}}$ of the next block.
We argue that the quality of speaker embeddings is critical for diarization performance. Therefore, we design a filter to specifically extract speaker embeddings suitable for diarization. The filter requires the speaker timestamps in the session, which are provided by the x-vector/AHC diarization step. With the speaker timestamps and the MFCCs of the session, the filter generates the corresponding speaker embeddings using ResNet and Attentive Statistic Pooling [34], as shown in Fig. 1. The output of the filter is called the z-vector. These z-vectors can be used as detector inputs instead of x-vectors or i-vectors. By integrating the filter with the MTFAD detector, speaker embedding extraction and detection is combined into an end-to-end process. In addition, since the filter and detector are trained jointly, the z-vector is expected to be more suitable for the diarization problem than the x-vector and i-vector.

### 3. Experiments

Two corpora were involved in our experiments: one is simulated from the Switchboard and NIST SRE datasets, and the other is CALLHOME. All overlapping regions were counted in the performance evaluation. The results were evaluated by diarization error rate (DER) and Jaccard error rate (JER). JER, introduced by DIHARD II, is based on the Jaccard index [35].

#### 3.1. SWB+SRE Simulated Corpus

We utilized the additional Switchboard corpus and the NIST SRE dataset. The total number of speakers in the 683 hours of data from SRE and Switchboard was 6,392. We simulated the training data by Algorithm 1 in [28]. To achieve an overlap ratio of 20%, with two, three, and four participants, we chose the ratio of 20%, with two, three, and four participants, we chose the ratio of 20%, with two, three, and four participants, we chose the ratio of 20%, with two, three, and four participants, we chose the ratio of 20%, with two, three, and four participants, we chose.

| Ratio (|T|/(|T|+|I|)) | Threshold | Oracle | Ideal |
|-----------------|-----------|--------|-------|
| 0%              | 29.92     | 26.37  | 16.65 |
| 25%             | 29.82     | 25.91  | 12.03 |
| 50%             | 30.05     | 26.26  | 11.14 |
| 75%             | 29.80     | 26.22  | 10.36 |
| 100%            | 30.43     | 27.74  | 10.22 |

#### 3.2. CALLHOME

CALLHOME, also known as NIST SRE 2000 CALLHOME (LDC2001S97), is a telephone dataset that contains conversations in multiple languages. A total of 500 conversations are recorded at a sampling rate of 8kHz. The number of speakers in each conversation varies from two to seven. Since the CALLHOME dataset was too small to train our model, we used the SWB+SRE dataset for pre-training.

In the training phase, we pre-trained the MTFAD models on the SWB+SRE dataset. Because there is no official development set defined in the CALLHOME task, we followed the instructions in the Kaldi recipe to split the set equally into two

| Method          | Threshold | Oracle |
|-----------------|-----------|--------|
|                 | DER       | JER    | DER   | JER    |
| x-vector/AHC    | 38.49     | 53.38  | 40.38 | 52.58  |
| MTFAD (i-vector)| 23.72     | 36.21  | 19.50 | 29.24  |
| MTFAD (x-vector)| 25.16     | 38.91  | 18.61 | 28.88  |
| MTFAD (z-vector)| 23.06     | 32.67  | 13.46 | 18.95  |

parameters β for three, six, and nine seconds in the algorithm, resulting in 137, 226 and 320 hours of data, respectively.

In the inference phase, we used x-vector/AHC to produce the first-stage RTTM files on the simulated set. Each speaker’s representation vector was extracted from a cascade of time frames in which only he or she was active. In both phases, the i-vector and x-vector speaker representations were obtained by Kaldi [36] pre-trained extractors. For the z-vector case, the first-stage RTTM was used as the input to the diarization models.
In this paper, we have proposed the MTFAD model, which is developed from a DPRNN-based detector and a z-vector filter, for speaker diarization. The DPRNN structure in the MTFAD detector allows the model to handle conversations with varying numbers of speakers and use data with any number of speakers in training. Currently, the performance bottleneck comes from the mis-prediction of the first-stage x-vector/AHC. We plan to retrofit the original MTFAD so that it can automatically correct the estimated number of speakers in a conversation provided by the first stage, or even drop the first stage. In addition, the current post-filter is a simple median filter, and we want to replace it with stronger filters such as VB and HMM. Furthermore, we would like to incorporate reconstruction losses commonly used in speech separation to enhance diarization.

4. Conclusions and Future Work

In this paper, we have compared the effects of different speaker embedding models, including z-vector, i-vector, and x-vector. From Table 4, we can see that all the three MTFAD models outperform not only the x-vector/AHC baseline but also the strong SA-EEND-EDA model [29]. Furthermore, the results again confirm that the z-vector-based MTFAD is better than the i-vector-based and x-vector-based MTFAD under both Threshold and Oracle conditions. We also notice that greater improvements are seen under the Oracle condition. This is because that when the number of speakers is correct in the first-stage diarization, the model can estimate a more accurate z-vector for each speaker. In contrast, under the Threshold condition, the wrong number of speakers predicted in the first stage x-vector/AHC may cause some z-vectors to not match the actual speakers, thus leading to small improvements in DER and JER. The results in Tables 3 and 4 show that with the well-designed filter and detector, the MTFAD model is a flexible and effective diarization model that can extract more accurate speaker vectors to handle conversations with different numbers of speakers.

Finally, we compared the proposed MTFAD model with more existing models. Since most end-to-end models were only evaluated in 2-speaker experiments, we compared different models in the 2-speaker CALLHOME task. Table 5 shows the results. It is clear that MTFAD outperforms all models, with a 12.2% relative reduction in DER over the baseline x-vector/AHC [29]. From Tables 4 and 5, we conclude that the MTFAD model outperforms all models compared in this paper in both 2-speaker and multi-speaker tasks.
5. References

[1] X. Anguera, S. Bozonnnet, N. Evans, C. Fredouille, and G. Friedrich, “Speaker diarization: A review of recent research,” IEEE Transactions on Audio, Speech and Language Processing, vol. 20, no. 2, pp. 356–370, 2012.

[2] N. Ryant, P. Singh, V. Krishnamohan, R. Varma, K. Church, C. Cieri, J. Du, S. Ganapathy, and M. Liberman, “The third DIHARD diarization challenge,” 2020. [Online]. Available: http://arxiv.org/abs/2012.01477

[3] M. Diels, L. Burget, and P. Matjejka, “Speaker diarization based on bayesian HMM with eigenvoice priors,” in Proc. Odyssey, 2018.

[4] Y. Fujita, N. Kanda, S. Horiguchi, K. Nagamatsu, and S. Watanabe, “End-to-end neural speaker diarization using permutation-free objectives,” in Proc. Interspeech, 2019.

[5] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, “X-vectors: Robust DNN embeddings for speaker recognition,” in Proc. ICASSP, 2018.

[6] Daniel Garcia-Romero, David Snyder, Gregory Sell, Daniel Povey, and Alan McCree, “Speaker diarization using deep neural network embeddings,” in Proc. ICASSP, 2017.

[7] M. Diels, L. Burget, S. Wang, J. Rohdin, and H. Černocký, “Bayesian HMM based x-vector clustering for speaker diarization,” in Proc. Interspeech, 2019.

[8] G. Sell, D. Snyder, A. Mccree, D. Garcia-Romero, J. Villalba, M. Maciejewski, V. Manohar, N. Dekh, D. Povey, S. Watanabe, and S. Khudanpur, “Diarization is hard: Some experiences and lessons learned for the JHU team in the inaugural DIHARD challenge,” in Proc. Interspeech, 2018.

[9] N. Dekh, F. J. Kenny, R. Dekh, P. Dumouchel, and P. Ouellet, “Front-end factor analysis for speaker verification,” IEEE Transactions on Audio, Speech and Language Processing, vol. 19, pp. 788–798, 2011.

[10] G. Sell and D. Garcia-Romero, “Speaker diarization with PLDA i-vector scoring and unsupervised calibration,” in Proc. IEEE SLT, 2014.

[11] L. Wan, Q. Wang, A. Papir, and I. L. Moreno, “Generalized end-to-end loss for speaker verification,” in Proc. ICASSP, 2018.

[12] Aohan Zhang, Quan Wang, Zhenyao Zhu, John Paisley, and Chong Wang, “Fully supervised speaker diarization,” in Proc. ICASSP, 2018.

[13] S. Ioffe, “Probabilistic linear discriminant analysis,” in Proc. ECCV, 2006.

[14] Simon J.D. Prince and James H. Elder, “Probabilistic linear discriminant analysis for inferences about identity,” in Proc. ICCV, 2007.

[15] P. Kenny, T. Stafylakis, P. Ouellet, M. Jahangir Alam, and P. Dumouchel, “PLDA for speaker verification with utterances of arbitrary duration,” in Proc. ICASSP, 2013.

[16] H. Ning, M. Liu, H. Tang, and T. Huang, “A spectral clustering approach to speaker diarization,” in Proc. Interspeech, 2006.

[17] T. J. Park, K. J. Han, J. Huang, X. He, B. Zhou, P. Georgiou, and S. Sarayanan, “Speaker diarization with lexical information,” in Proc. ICASSP, 2019.

[18] C. Fredouille, S. Bozonnet, and N. Evans, “The LIA-EURECOM RT’09 speaker diarization system,” in Proc. RT, 2009.

[19] S. H. Shum, N. Dekh, R. Dekh, and J. R. Glass, “Unsupervised methods for speaker diarization: An integrated and iterative approach,” IEEE Transactions on Audio, Speech and Language Processing, vol. 21, no. 10, pp. 2015–2028, 2013.

[20] G. Sell and D. Garcia-Romero, “Diarization resegmentation in the factor analysis subspace,” in Proc. ICASSP, 2015.

[21] M. Sahidullah, J. Patino, S. Cornell, R. Yin, S. Sivasankaran, H. Bredin, P. Korshunov, A. Brütii, R. Serizel, E. Vincent, N. Evans, S. Marcel, S. Squartini, and C. Barras, “The speed submission to DIHARD II: Contributions & lessons learned,” 2019. [Online]. Available: http://arxiv.org/abs/1911.02388

[22] Y. Fujita, S. Watanabe, S. Horiguchi, Y. Xue, and K. Nagamatsu, “End-to-End neural diarization: Reformulating speaker diarization as simple multi-label classification,” 2020. [Online]. Available: http://arxiv.org/abs/2003.02966

[23] Thilo von Neumann, Keisuke Kinoshita, Marc Delcroix, Shoko Araki, Tomohiro Nakatani, and Reinhold Haeb-Umbach, “All-neural online source separation, counting, and diarization for meeting analysis,” in Proc. ICASSP, 2018.

[24] 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.

[25] Latané Bullock, Hervé Bredin, and Leibny Paola Garcia-Perera, “Overlap-aware diarization: Resegmentation using neural end-to-end overlapped speech detection,” in Proc. ICASSP, 2020.

[26] Dang Yu, Morten Kolbak, Zheng-Hua Tan, and Jesper Jensen, “Permutation invariant training of deep models for speaker-independent multi-talker speech separation,” in Proc. ICASSP, 2017.

[27] N. Zeghidour, O. Teboul, and D. Grangier, “DIVE: End-to-end speech diarization via iterative speaker embedding,” 2021. [Online]. Available: http://arxiv.org/abs/2105.13802

[28] 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 ASRU, 2019.

[29] S. Horiguchi, Y. Fujita, S. Watanabe, Y. Xue, and K. Nagamatsu, “End-to-end speaker diarization for an unknown number of speakers with encoder-decoder based attractors,” in Proc. Interspeech, 2020.

[30] S. Horiguchi, P. Garcia, Y. Fujita, S. Watanabe, and K. Nagamatsu, “End-to-end speaker diarization as post-processing,” in Proc. ICASSP, 2021.

[31] I. Medennikov, M. Korenevsky, T. Psiyach, Y. Khokhlov, M. Korenevskaya, I. Sorokin, T. Timofeeva, A. Mitrofanov, A. Andrusenko, I. Podluzhny, A. Laptev, and A. Romanenko, “Target-speaker voice activity detection: A novel approach for multi-speaker diarization in a dinner party scenario,” in Proc. Interspeech, 2020.

[32] Yi Luo, Zhuo Chen, and Takuya Yoshioka, “Dual-path RNN: Efficient long sequence modeling for time-domain single-channel speech separation,” in Proc. ICASSP, 2020.

[33] C. Li, Y. Luo, C. Han, J. Li, T. Yoshioka, T. Zhou, M. Delcroix, K. Kinoshita, C. Boedeker, Y. Qian, S. Watanabe, and Z. Chen, “Dual-path RNN for long recording speech separation,” in Proc. IEEE SLT, 2021.

[34] K. Okabe, T. Koshinaka, and K. Shinoda, “Attentive statistics pooling for deep speaker embedding,” Proc. Interspeech, 2018.

[35] “Jaccard index.” [Online]. Available: https://en.wikipedia.org/wiki/Jaccard_index

[36] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlíček, Y. Qian, P. Schwarz, J. S. Silovsky, G. Stemmer, and K. V. Vesely, “The Kaldi speech recognition toolkit,” in Proc. IEEE ASRU, 2011.