Sequence-Level Speaker Change Detection With Difference-Based Continuous Integrate-and-Fire

Zhiyun Fan, Linhao Dong, Meng Cai, Zejun Ma, and Bo Xu

***Abstract*—Speaker change detection is an important task in multi-party interactions such as meetings and conversations. In this paper, we address the speaker change detection task from the perspective of sequence transduction. Specifically, we propose a novel encoder-decoder framework that directly converts the input feature sequence to the speaker identity sequence. The difference-based continuous integrate-and-fire mechanism is designed to support this framework. It detects speaker changes by integrating the speaker difference between the encoder outputs frame-by-frame and transfers encoder outputs to segment-level speaker embeddings according to the detected speaker changes. The whole framework is supervised by the speaker identity sequence, a weaker label than the precise speaker change points. The experiments on the AMI and DIHARD-I corpora show that our sequence-level method consistently outperforms a strong frame-level baseline that uses the precise speaker change labels.**

***Index Terms*—Difference-based continuous integrate-and-fire, sequence transduction, speaker change detection.**

I. Introduction

MULTI-PARTY interactions such as meetings and conversations are one of the most important scenarios for many speech and language applications [1]. Speaker change detection (SCD), the task of finding the time points that a new speaker starts to speak, is critical for such applications and has received increasing attention in recent years [2]–[5].

SCD is known as an important part of speaker diarization [6], [7]. It was previously modeled with distance-based methods [3], [8], [9], which segment audio with a sliding window, and the distance of speaker embedding is used to decide whether a speaker change happens between the adjacent segments. Since the pitch varies saliently with speaker changes, some pitch-based methods detect speaker changes with the change in pitch [10]–[12]. More recently, there are some attempts at predicting the speaker change at the end of the neural network without relying on a distance metric [2], [4], [13], [14]. Almost all these end-to-end systems are based on binary classification (i.e. change or not) to predict whether a speaker change happens between frames or segments. Since these methods rely on the precise speaker change labels, they are categorized into frame-level models.

Recently, the sequence-level modeling methods have made great progress in automatic speech recognition (ASR) [15]–[18]. These models rely on different alignment mechanisms to conduct sequence transduction and have shown their performance advantages in comparison with the frame-level hybrid models [19]–[21]. The success of the sequence-level model in ASR inspires us that it may be suitable for the SCD task.

In this paper, we model the SCD task from the perspective of sequence transduction. Specifically, we propose a novel encoder-decoder model to convert the input feature sequence to the speaker identity sequence. Inspired by the success of Continuous Integrate-and-fire (CIF) in the ASR field [22], we design a difference-based continuous integrate-and-fire (DCIF) mechanism to bridge the encoder and decoder. The DCIF performs two functions in the SCD task, including 1) detecting the speaker changes and splitting the encoded sequence into segments according to the SCD results, 2) calculating segment-level speaker embeddings and firing them to the decoder. Then the decoder predicts the speaker identity. Based on the above framework, our method processes on the sequence level and removes the need for frame-level speaker change labels in the training. Besides providing the model framework, we also present several effective methods used to complete our sequence-level model, including 1) a DifferNet to estimate the speaker difference for the DCIF, 2) the length normalization to better represent the fired speaker embeddings, and 3) a multi-label focal loss (MLFL) to boost the training.

We evaluate our sequence-level model on a real recording meeting corpus, AMI [23] and DIHARD-I corpus [24]. After exploring three important model settings, our method achieves 86.76% and 89.29% harmonic mean (Hn) of purity and coverage on AMI and DIHARD-I, respectively, outperforming the Hn of 86.00% and 88.09% from a strong frame-level baseline [7]. In addition, we provide the ablation study to evaluate the importance of the applied methods. Our contributions are summarized as follows: 1) As far as we know, we are the first to address the SCD task as sequence transduction and propose a sequence-level SCD framework. 2) We design a DCIF mechanism to detect speaker changes and automatically calculate segment-level speaker embeddings according to the detected results. 3) We demonstrate that our sequence-level method achieves a better SCD performance with weaker supervision than a strong frame-level SCD baseline and release our code at https://github.com/zhiyunfan/SEQ-SCD.

II. Related Work

Most previous SCD methods [2]–[4], [8], [9], [13], [14] detect speaker changes between frames or fixed-size windows by thresholding the distance or binary classification. These methods rely on the precise speaker change labels during training and are categorized into frame-level methods. In contrast, the sequence-level SCD first proposed in this paper addresses the SCD task as sequence transduction, which transfers the input feature sequence to the speaker identification sequence and predicts speaker changes by the specially designed DCIF. Benefit from the novel sequence-level model structure, our training process gets rid of the dependence on precise speaker change labels used in the frame-level methods.

Manuscript received 14 April 2022; revised 4 June 2022; accepted 16 June 2022. Date of publication 24 June 2022; date of current version 19 July 2022. This work was supported in part by the National Innovation 2030 Major S&T Project of China under Grant 2020AAA0104202, in part by the Key Research Program of the Chinese Academy of Sciences under Grant ZDBS-SSW-JSC006, and in part by the Strategic Priority Research Program of the Chinese Academy of Sciences under Grant XDA27030300. The associate editor coordinating the review of this manuscript and approving it for publication was Mr. Ville M. Hautamaki. *(Zhiyun Fan and Linhao Dong contributed equally to this work.) (Corresponding author: Zhiyun Fan.)*

Zhiyun Fan and Bo Xu are with the Chinese Academy of Sciences Institute of Automation, Beijing 100190, China (e-mail: fanzhiyun2017@ia.ac.cn; xubo@ia.ac.cn).

Linhao Dong, Meng Cai, and Zejun Ma are with the Beijing Bytedance Technology Company Ltd, Beijing 100098, China (e-mail: donglinhao@bytedance.com; caimeng@bytedance.com; mazejun@bytedance.com).

Digital Object Identifier 10.1109/LSP.2022.3185955

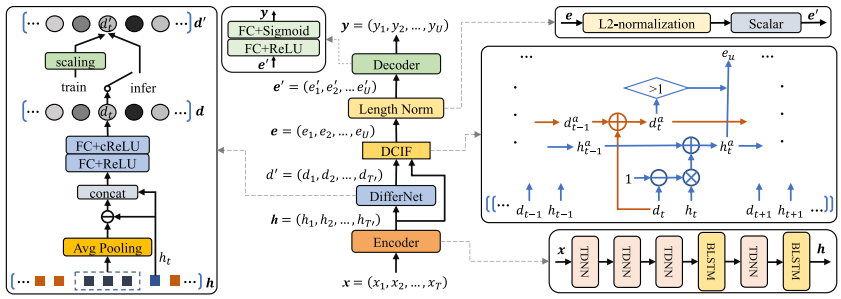

Fig. 1. The architecture of sequence-level SCD model. The main body is in the middle of the diagram, and the details of the DifferNet, the DCIF, the length normalization layer and the encoder are shown in the boxes on both sides.

Our DCIF is inspired by the CIF in a sequence-level ASR model [22]. The CIF uses a pre-computed weight that scales the frame-level acoustic information contained in each frame to weight the frame-level representation and generates label-level representations. Different from the CIF, the DCIF integrates pre-computed speaker difference between each frame and its corresponding context instead of speaker information contained in each frame, and transfers frame-level speaker representations to segment-level speaker representations. Within the transfer, the weights of each frame and speaker difference are negatively correlated.

III. Speaker Change Detection as a Sequence Transduction Task

A. Sequence Transduction

We address SCD as a sequence transduction task. The input is a sequence of features, $x = (x_1, x_2, \ldots, x_T)$, where T is the total length of the input sequence. The output is a sequence of speaker identities $\boldsymbol{y} = (y_1, y_2, \ldots, y_U)$, where U is the number of segments of the input sequence after partitioning it on speaker change boundaries. Thus $U-1$ is the number of speaker change points. An encoder-decoder framework connected by a dynamic segmentation module is used for the sequence transduction of SCD. The encoder transforms the input sequence to the frame-level speaker representations. The dynamic segmentation module detects speaker changes and splits the encoder outputs into segments according to the detected speaker change points. The speaker embeddings of the split segments are sent to the decoder for speaker classification. The objective is to find a function $f : x \rightarrow y$ that transforms the input feature sequence into a speaker identity sequence. The whole framework can be optimized as follow:

$$L = \frac{1}{U}\sum_{u=1}^{U} \text{Classification\_loss}(f(x)_u, y_u) \tag{1}$$

B. DCIF in the Sequence Transduction of SCD

We propose a novel DCIF mechanism suitable for the SCD task to conduct the sequence transduction in Section III-A. It forwardly accumulates the speaker difference and integrates the speaker embedding simultaneously. Once the accumulated speaker difference reaches a threshold, the integrated speaker embedding will be fired for further speaker classification.

The sequence-level SCD model is shown in Fig. 1. The encoder and the decoder are connected by the DCIF. The DifferNet and the length normalization layer are designed to cooperate with the DCIF. Specifically, the encoder transforms the input feature sequence $\boldsymbol{x} = (x_1, x_2, \ldots, x_T)$ to the frame-level speaker embedding $\boldsymbol{h} = (h_1, h_2, \ldots, h_{T'})$. The DifferNet predicts the speaker difference $\boldsymbol{d'} = (d'_1, d'_2, \ldots, d'_{T'})$, each of which is a scalar and corresponds to each encoded frame. The DCIF receives the frame-level speaker embedding $\boldsymbol{h}$ and the speaker difference $\boldsymbol{d'}$. Then it forwardly accumulates the speaker difference d^a_t and integrates the speaker embedding h^a_t, which is an accumulation sum of h_t weighted by $1 - d'_t$. Once the accumulated speaker difference value reaches a threshold β, a speaker change point is located. Then the current speaker difference value will be divided into two parts, one for completing the current integration and the other for the next integration, and the currently integrated speaker embedding will be fired for further speaker classification. Until the last frame, we save the currently integrated speaker embedding for the last speaker. More details are shown in Algorithm 1.

Algorithm 1: DCIF in the Sequence-Level SCD.

Input: The encoded frames $\boldsymbol{h} = (h_1, h_2, \ldots, h_{T'})$, the speaker difference value $\boldsymbol{d'} = (d'_1, d'_2, \ldots, d'_{T'})$, the threshold β;
Output: The fired speaker embeddings $\boldsymbol{e} = (e_1, e_2, \ldots, e_U)$;

```
Initialize u=1, the accumulated speaker difference value
  d^a_0=0, the accumulated speaker embedding state h^a_0 = h_1;
for t = 1; t <= T'; t++ do
  // calculate currently accumulated speaker difference value,
    and integrated speaker embedding state;
  d^a_t = d^a_{t-1} + d'_t;
  h^a_t = h^a_{t-1} + (1 - d'_t) * h_t;
  if d^a_t > β then // a speaker change is detected
    // firing currently integrated speaker embedding state
    e_u = h^a_t;  u++;
    h^a_t = h_t; // reset integrated speaker embedding state;
    // d'_t is divided into two part, the second part d'_{t2} is used
    to reset accumulated speaker difference;
    d'_{t1} = 1 - d^a_{t-1};  d'_{t2} = d'_t - d'_{t1};
    d^a_t = d'_{t2};
  end if
end for
e_U = h^a_t; //save the speaker embedding for the last speaker;
return e = (e_1, e_2, ..., e_U);
```

In the inference stage, we save a mark sequence $c = [c_t \in \{0, 1\} | t = 1, \ldots, T']$ along with the calculation of the DCIF. The $c_t = 1$ indicates that the accumulated difference reaches the threshold at the t-th time step. The mark sequence is used for the calculation of SCD metrics.

IV. Model Details

To boost the performance of our sequence-level SCD, we propose the following three methods:

DifferNet: As shown in Fig. 1, the DifferNet receives the frame-level speaker embedding h_t and predicts speaker difference value d'_t for each encoded frame. The speaker difference d'_t is determined by h_t and its corresponding history chunk.

$$o_1 = h_t - \frac{1}{l}\sum_{\tau=t-l}^{t-1} h_\tau \tag{2}$$

$$o_2 = W_2 * \text{ReLU}(W_1 * [o_1; h_t] + b_1) + b_2 \tag{3}$$

$$d_t = \min(\max(o_2, 0), 1) \tag{4}$$

where l is the length of the history chunk. o_1 and h_t are concatenated and fed into the two FC layers. h_t is used to reduce the interference of silence and noise in the measure of speaker difference d'_t. W_1, W_2, b_1, b_2 are trainable parameters. Equation 4 is the formulation of cReLU [25] with an upper bound 1. After finishing the calculation of speaker difference value d_t for all encoded frames, the scaling operation is applied during

training.

$$d'_t = \kappa * d_t \tag{5}$$

$$\kappa = (U-1)/\sum_{t=1}^{T} d_t \tag{6}$$

where U is the length of the speaker identity sequence. The scaling operation ensures that the sum of $\boldsymbol{d}'$ is equal to the number of speaker changes $U-1$, which could make the number of fired speaker embeddings equal to the length of the speaker identity sequence. In the inference stage, the scaling operation is not used, which means that d'_t is identical to d_t.

Length Normalization: The speaker embedding e_u fired by the DCIF is a weighted sum of a varying number of frames (e_u is detailed in Algorithm 1). We use an L2-normalization layer followed by a scalar [26] to normalize the speaker embedding into a fixed hyperspace. The normalized embedding e'_u is euqal to $\eta * e_u/||e_u||_2$. η is a hyper-parameter used to scale the unit-length speaker embedding into a fixed radius.

Loss Function: The loss function is the interpolation of a multi-label focal loss (MLFL) and a quantity loss [22].

$$L = \lambda_1 \frac{1}{U}\sum_u \mathrm{MLFL}(p_u, y_u) + \lambda_2 |U - 1 - \sum_t d'_t| \tag{7}$$

$$\mathrm{MLFL}(p_u, y_u) = \frac{1}{C}\sum_c (-\alpha(1-p_{u,c})^\gamma y_{u,c}\log(p_{u,c}) - (1-\alpha)p_{u,c}^\gamma (1-y_{u,c})\log(1-p_{u,c})) \tag{8}$$

where $y_u = [y_{u,c} \in \{0,1\} | c = 1, \ldots, C]$, and $y_{u,c} = 1$ indicates that the speaker c is presenting at segment u. The p_u predicted by the decoder is the element-wise sigmoid activation for the C speakers. MLFL is a combination of binary cross-entropy (BCE) [27] loss and focal loss [28]. Since y_u may contain multiple speakers, we choose the BCE loss rather than the softmax. The focal loss makes the model focus on the positive samples and down-weight the numerous negative samples. The α and γ are two hyper-parameters.

The second item is a quantity loss which promotes the predicted firing times of the DCIF closer to the target number of speaker change points $U-1$. The λ_1 and λ_2 are two tunable hyper-parameters.

V. Experiments and Results

A. Experimental Setup

Experiments are performed on AMI [23] and DIHARD-I corpus [24]. The AMI is a real-recorded 100-hour English meeting corpus. We use Mix-Headset recordings for our experiments, and the division of the AMI corpus is consistent with the baseline system [29]. For DIHARD-I corpus, we split the development set into two parts: 131 files used as training set and the remaining 33 files used as a new development set. The new development set is simply referred to development set in the following. We share the split at https://github.com/zhiyunfan/SEQ-SCD/tree/master/data/dihard1.

For the model structure, the encoder stacks four Time Delay Neural Network (TDNN) layers and two Bi-LSTM layers. The details are shown in the lower right corner of Fig. 1. The two Bi-LSTM layers both have 256 hidden units. The four TDNN layers have 512 channels with the context of [−2,−1,0,1,2], which sums up to five frames. The strides of the TDNN layers change with the number of temporal downsampling. (1,1,1,1), (1,1,1,2), (1,1,2,2), (1,2,2,2) and (2,2,2,2) is for $1/1$, $1/2$, $1/4$, $1/8$, $1/16$ downsampling, respectively. In the DifferNet, the length of the history chunk is explored in Section V-B. The two FC layers are 512- and 1-dimensional, respectively. The hyper-parameter η follows the best value 12 in [26]. For the DCIF, we set the β to 1.0. The decoder consists of two FC layers. The hidden layer has 256 units with ReLU activation. The output layer has 136 (the number of speakers in the training set) units with sigmoid activation. The loss hyper-parameters λ_1 and λ_2 is set to 50.0 and 1.0. The α and γ in the MLFL are set to 0.25 and 2, the best value given by Lin *et al.* [28].

TABLE I
Evaluation of the Sequence-Level SCD on the Development Set of AMI With Various Model Settings

		Purity	Coverage	Hn
Size of window	1 s	-	-	-
	2 s	77.39	90.79	83.55
	4 s	81.39	87.41	**84.29**
	6 s	78.18	87.03	82.37
	8 s	75.31	88.91	81.55
Down-sampling	1/1	81.39	87.41	84.29
	1/2	80.77	88.04	84.25
	1/4	82.65	87.17	84.85
	1/8	82.69	87.91	**85.22**
	1/16	81.95	87.26	84.52
Length of history	80 ms	82.87	87.21	84.99
	160 ms	82.69	87.91	**85.22**
	240 ms	80.80	89.70	85.02
	320 ms	81.58	88.69	84.99
	400 ms	80.43	90.03	84.96

For the online processing, the input batch is randomly sampled from the raw session audio with a window. Then we apply additive noise from MUSAN dataset [30] on-the-fly. The SNR values are sampled from 5 to 20 dBs. Specially, reverberation noise is used in Section V-C. The room size is ranging from 2 m-1 m-2 m to 10 m-10 m-5 m (length-width-height). The wall absorption coefficient is sampled from 0.2 to 0.9. We extract 59-dimensional MFCC features (19 coefficients and energy with first- and second-order derivatives) with 25 ms frame length and 10 ms frame shift. The batch size is 128, and the length of the window is explored in Section V-B. We use Adam [31] optimizer, warming up the learning rate for the first 5% of updates to a peak of 10^{-4}, and holding on for the next 50%, and then linearly decaying for the remainder.

During inference, the metrics of all models follow the tool of Pyannote [7]. Firstly, we split each long test audio into fixed-length segments as same as training. And there is an 80% overlap between two adjacent segments. For each frame, the final speaker change score is the average result of all segments containing this frame. Then the frames corresponding to prediction scores which are local maxima and greater than a tunable threshold θ are marked as speaker change points. All our experiments are evaluated on the purity, coverage [7] and their harmonic mean (Hn). The tunable threshold is tuned on the development set to maximize the Hn.

B. Exploration on Model Settings

Firstly, we explore three model settings in our sequence-level SCD model, including the size of the window used to sample batch, the temporal down-sampling in the encoder and the length of the history chunk used to calculate the speaker difference value. The size of the window affects the number of speaker change points in the batch. The temporal down-sampling decides the length of the encoded frame sequence fed into the DCIF. The length of the history chunk directly affects the calculation of speaker difference value.

Table I shows the results of our sequence-level model with various settings on the development set of AMI. In the upper part, we investigate the size of the window used to sample the batch. We fix a $1/1$ temporal down-sampling of the encoder and 160 ms history chunk. As can be seen, the 4 s window achieves the best Hn value. Compared with the 2 s window, the 4 s window provides more speaker changes during training. But the performance degrades when the size of the window keeps increasing. Then we try to reduce the length through the temporal down-sampling of the encoder with a 4 s window and 160 ms history chunk. The results of various temporal down-sampling are shown in the middle of Table I. As the temporal down-sampling increases, our model obtains further performance gains. The $1/8$ temporal down-sampling gets the best results. The performance degradation of the $1/16$ may be due to multiple speaker change points covered by one encoded frame (There are a large number of rapid speaker change points in the AMI corpus). Finally, we fix a 4 s window and $1/8$ temporal down-sampling to compare the various length of the history chunk. The results are shown in the bottom part of Table I. We find that the Hn value slightly fluctuates with the changing of the length of the history chunk, and the 160 ms history chunk achieves the best performance. In summary,

TABLE II
EVALUATION OF THE BASELINE MODEL AND OUR BEST MODEL ON THE TEST SET OF AMI AND DIHARD-I CORPORA

	AMI			DIHARD-I		
	Purity	Coverage	Hn	Purity	Coverage	Hn
Pyannote [29]	83.00	89.30	86.00	84.99	91.43	88.09
Ours	**83.92**	**89.81**	**86.76**	**86.24**	**92.56**	**89.29**

TABLE III
EVALUATION OF THE ABLATION STUDY ON THE DEVELOPMENT AND TEST SET OF AMI. RESULTS ON THE DEVELOPMENT SET USE A SMALL FONT

	Purity	Coverage	Hn
Full model	83.92 82.65	89.81 88.56	86.76 85.50
w/o Length Norm	81.66 79.91	91.24 90.24	86.18 84.76
w/o Scaling	83.87 82.66	88.33 87.27	86.05 84.90
w/o Focal Loss	82.41 81.97	87.97 86.29	85.10 84.08

for our sequence-level SCD model, 4 s window to sample batch, $1/8$ temporal down-sampling, and 160 ms history chunk are relatively better model settings, which will be used for the subsequent experiments.

C. Comparison With Baseline

In this section, we compare our model with the baseline model on the AMI and DIHARD-I corpora. The baseline results are achieved in an open-source toolkit, Pyannote [29]. It directly predicts frame-level SCD results and applies the same Bi-LSTM layer as our model to conduct binary sequence labeling. Considering that the baseline model was trained for 1000 epochs, we increase the training epoch of our model from 160 to 500. The results in Table II show that our sequence-level model consistently outperforms the baseline model on the two corpora. In addition, we compare our method with the baseline system on AMI corpus adding reverberation noise. The baseline and our method achieve 83.36%-88.86%-86.02% and 84.25%-89.83%-87.01% (Purity-Coverage-Hn), respectively. It reflects the robustness of the proposed sequence-level SCD method.

D. Ablation Study

In this section, we use the ablation study to evaluate the importance of different methods applied to the sequence-level SCD model. As shown in Table III, the first row is the results achieved by the full model. In the following three experiments, we ablate the length normalization, the scaling operation and the focal loss, respectively. The results indicate that all three methods provide improvements. Among them, ablating the focal loss causes the largest performance degradation, which indicates that the focal loss alleviates the imbalance of positive and negative samples as we expected.

VI. CONCLUSION

In this paper, we address the speaker change detection task from the perspective of sequence transduction and propose a sequence-level SCD model using difference-based continuous integrate-and-fire (DCIF). Evaluated on the AMI and DIHARD-I corpora, our proposed sequence-level model achieves 86.76% and 89.29% harmonic mean (Hn) of purity and coverage without using any precise frame-level speaker change label, and outperforms the 86.00% and 88.09% Hn from a strong frame-level baseline [7]. It demonstrates the effectiveness of the sequence-level model in the SCD task.

REFERENCES

[1] G. Sun, C. Zhang, and P. C. Woodland, "Combination of deep speaker embeddings for diarisation," *Neural Netw.*, vol. 141, pp. 372–384, 2021.

[2] R. Yin, H. Bredin, and C. Barras, "Speaker change detection in broadcast tv using bidirectional long short-term memory networks," in *Proc. Int. Speech Commun. Assoc.*, 2017, pp. 3827–3831.

[3] Z. Ge, A. N. Iyer, S. Cheluvaraja, and A. Ganapathiraju, "Speaker change detection using features through a neural network speaker classifier," in *Proc. Intell. Syst. Conf.*, 2017, pp. 1111–1116.

[4] M. Hrúz and Z. Zajíc, "Convolutional neural network for speaker change detection in telephone speaker diarization system," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2017, pp. 4945–4949.

[5] L. Sari, M. Hasegawa-Johnson, and S. Thomas, "Auxiliary networks for joint speaker adaptation and speaker change detection," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 29, pp. 324–333, 2021.

[6] 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.

[7] H. Bredin, "pyannote.metrics: A toolkit for reproducible evaluation, diagnostic, and error analysis of speaker diarization systems," in *Proc. Int. Speech Commun. Assoc.*, 2017, pp. 3587–3591.

[8] X. Anguera, S. Bozonnet, N. Evans, C. Fredouille, G. Friedland, and O. Vinyals, "Speaker diarization: A review of recent research," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 20, no. 2, pp. 356–370, Feb. 2012.

[9] S. Chen *et al.*, "Speaker, environment and channel change detection and clustering via the Bayesian information criterion," in *Proc. Broadcast News Transcription Understanding Workshop*, 1998, pp. 127–132.

[10] M. Yang, Y. Yang, and Z. Wu, "A pitch-based rapid speech segmentation for speaker indexing," in *Proc. Int. Symp. Multimedia*, 2005, pp. 571–576.

[11] B. Abdolali and H. Sameti, "A novel method for speech segmentation based on speakers' characteristics," *Signal & Image Process.*, vol. 3, no. 2, p. 65, 2012.

[12] A. O. Hogg, C. Evers, and P. A. Naylor, "Speaker change detection using fundamental frequency with application to multi-talker segmentation," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2019, pp. 5826–5830.

[13] S. H. Yella, A. Stolcke, and M. Slaney, "Artificial neural network features for speaker diarization," in *Proc. IEEE Spoken Lang. Technol. Workshop*, 2014, pp. 402–406.

[14] L. Sarı, S. Thomas, M. Hasegawa-Johnson, and M. Picheny, "Pre-training of speaker embeddings for low-latency speaker change detection in broadcast news," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2019, pp. 6286–6290.

[15] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2006, pp. 369–376.

[16] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel, and Y. Bengio, "End-to-end attention-based large vocabulary speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2016, pp. 4945–4949.

[17] A. Vaswani *et al.*, "Attention is all you need," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.

[18] A. Graves, "Sequence transduction with recurrent neural networks," 2012, *arXiv:1211.3711*.

[19] C.-C. Chiu *et al.*, "State-of-the-art speech recognition with sequence-to-sequence models," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2018, pp. 4774–4778.

[20] Y. He *et al.*, "Streaming end-to-end speech recognition for mobile devices," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2019, pp. 6381–6385.

[21] A. Gulati *et al.*, "Conformer: Convolution-augmented transformer for speech recognition," in *Proc. Int. Speech Commun. Assoc.*, 2020, pp. 5036–5040.

[22] L. Dong and B. Xu, "CIF: Continuous integrate-and-fire for end-to-end speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2020, pp. 6079–6083.

[23] J. Carletta, "Unleashing the killer corpus: Experiences in creating the multi-everything AMI meeting corpus," *Lang. Resour. Eval.*, vol. 41, pp. 181–190, 2007.

[24] N. Ryant *et al.*, "First dihard challenge evaluation plan," *Tech. Rep.*, 2018. [Online]. Available: https://zenodo.org/record/1199638

[25] J. Choi *et al.*, "PACT: Parameterized clipping activation for quantized neural networks," 2018, *arXiv:1805.06085*.

[26] W. Cai, J. Chen, and M. Li, "Analysis of length normalization in end-to-end speaker verification system," in *Proc. Int. Speech Commun. Assoc.*, 2018, pp. 3618–3622.

[27] Y. Fujita, N. Kanda, S. Horiguchi, K. Nagamatsu, and S. Watanabe, "End-to-end neural speaker diarization with permutation-free objectives," in *Proc. Int. Speech Commun. Assoc.*, 2019, pp. 4300–4304.

[28] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. Int. Conf. Comput. Vis.*, 2017, pp. 2980–2988.

[29] H. Bredin *et al.*, "Pyannote.audio: Neural building blocks for speaker diarization," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2020, pp. 7124–7128.

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

[31] D. P. Kingma and J. Ba, "ADAM: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Representations*, 2015, pp. 1–15.