Spectral Clustering Based EEND-vector Clustering: A Robust System Fine-tuned on Simulated Conversations

Kai Li

Shanghai Voicecomm Information Technology Co.,Ltd, China

lk9171@gmail.com

Abstract

In this paper, we propose an EEND-vector clustering based speaker diarization system we implemented in ISCSLP 2022 CSSD Challenge. Recently proposed EEND-vector clustering integrated the advantages of EEND-based and clustering-based diarization. In contrast to typical clustering-based systems, it considerably simplifies operations while being able to handle overlapping speech. In order to get a competitive result, we investigated several aspects to improve the EEND-vector clustering pipeline. (1) We investigated methods to create natural simulated conversation speech data, which is proven has significant impact on diarization result; (2) In order to capture the fine-gained local features, we introduced Conformer as the encoder in place of Transformer; (3) We modified the PIT loss function by adding an additive margin penalty to it; (4) We do model average by choosing the top 5 ranked models; (5) We use Spectral Clustering replace of AHC to do the embedding cluster and speaker count estimation. On MagicData-RAMC dataset with the CDER metric, our system achieved 21.9% on the Dev set and 24.5% on the Test respectively. In the ISCSLP 2022 CSSD Challenge, our system came in second with 8.1% CDER.

Index Terms: Speaker Diarization, EEND, Simulated Conversation, Spectral Clustering

1. Introduction

Speaker Diarization also known as speaker segmentation and clustering, is a technique of automatically dividing a conversation into parts based on speaker identification. It solved the "who-speaks-when" problem and is an essential pre-processing in automatic conversation analysis. With the result, post-processing like ASR can be promoted significantly. The conventional approaches, commonly known as the cluster-based approaches[1,2,3], typically first extract speech as short segments with VAD or SCD. Then, from frames, the embeddings representing speaker features are obtained. Finally, the embedding vectors would be clustered and regrouped according to speaker by applying an unsupervised clustering algorithm. Besides the complex modules and phases. The conventional methods also have trouble handling speaker overlaps. Compared to conventional approaches, the End-to-End Neural Diarization (EEND) methods[4,5,6,7] can handle multiple active speakers simultaneously. Thus have lately received a lot of research attention.

In comparison to the segment and cluster approaches, EEND is comparatively straightforward. Benefiting from the PIT[8], the EEND system can manage overlapping situations that the traditional approach cannot. In [4], the speaker diarization problem was treated as a multi-label classification problem. With a permutation-free objective it can estimate joint speech activities of all speakers frame-by-frame. SA-EEND[5] is a modified EEND system. It applied a self-attention mechanism to the EEND and begins outperforming conventional clustering-based methods. By introducing the encoder-decoder based attractor (EEND-EDA)[6] and a speaker-wise chain rule (SC-EEND)[7] to EEND, it is capable of handling situations when the speaker number is various. With its advantages, EEND became a new trend in solving speaker diarization problems.

Although EEND has lots of advantages, compared with the clustering-based methods it was experimentally shown to still have some shortcomings. Consider the memory consumption, EEND method generally designed to process the input audio chunk by chunk. This would decrease its accuracy while it dealing with long-form audio (e.g., duration greater than 10 min.). To solve this problem, EEND-vector clustering[9,10] the approach integrated the advantages of EEND and clustering into one was proposed. This kind of approach usually has two-stage, first it processes the audio with neural-network-based model chunk by chunk to get a label sequence and corresponding embedding vector. After that an unsupervised clustering algorithm would be applied to determine speaker correspondence among chunks and solve the inter-block label permutation problem. Benefit from the combination, the EEND-vector clustering framework[9] outperformed both vanilla EEND and x-vector clustering on simulated two-person conversation data with overlapping situations

Announced to have the ability to deal with long recordings. Those methods, however, only partially fixed some issues. Because of the chunk-wise process method, it inevitably has some limitations, especially while the audio is extremely long (e.g., duration greater than 30 min). As audio duration goes longer, the chunks number also increased. In [10], an agglomerative hierarchical clustering (AHC) algorithm was used to do the second stage post-processing. This algorithm starts by calculating the dissimilarity between objects and using a agglomeration criterion to control whether group two objects together or not. To implement the method, the agglomeration criterion is treated as a hyperparameter. This would decrease the robustness of the system when the chunks number is large, which is typical in the MagicData-RAMC dataset and many real world conversations.

In the ISCSLP 2022 CSSD Challenge, based on the EEND-vector clustering framework, we carefully analyzed each stage of the workflow and made some modifications to it. In this paper, we will elaborate details of these techniques used in our system, mainly from five aspects. Include: method of simulating conversation data; Conformer as the encoder of network; modification on PIT loss function; model averaging; Finally, the clustering algorithm. The remainder of the paper is organized as follows. In section 2 we review the original EEND-vector clustering framework. And then in section 3 we introduce the techniques we used in details. Finally, with experiments, we show how our system outperforms the baseline in ISCSLP 2022 CSSD Challenge.

2. EEND-vector Clustering Review

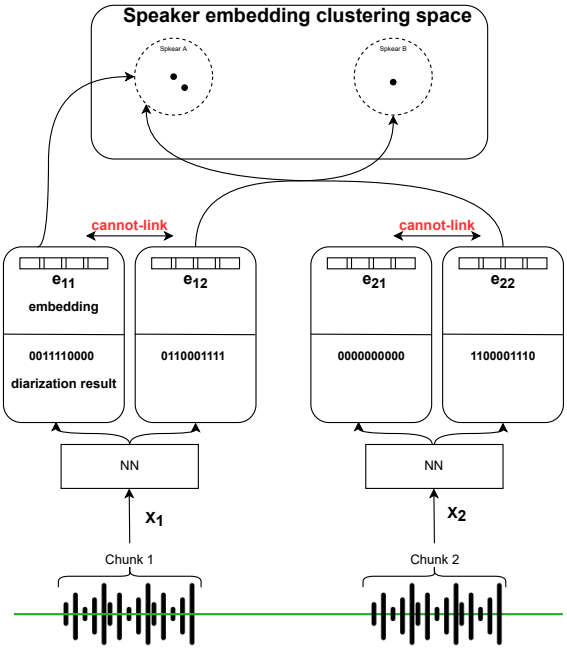

Figure 1: *Schematic diagram of the EEND-vector clustering framework.*

2.1. Overall framework

We cited Figure 1 from [10], which shows the overall framework of the EEND-vector clustering system.

As we can see from Figure 1, the system has two stages. At the first stage, input would be segmented into chunks and a sequence of the input frame features which represents the input chunk would be extracted, as $X_i = (x_{t,i} \mid t = 1, ..., T)$ where i, t and T are the chunk index, the frame index in the chunk and the chunk size respectively. $x_{t,t} \in R^K$ is a K dimensional input frame feature of frame t. Then, the chunk features are sent to the encoder, the output is the speaker diarization results and speaker embedding vectors. In this example, we assume that $S_Local = 2$, which means each chunk only has 2 speakers as max.

As aforementioned, consider of the memory limitation, the EEND system operates in a chunk-wise manner. Which is unavoidable has an inter-block label permutation issue. The second stage of EEND focuses on how to tackle this challenge. Commonly, an unsupervised clustering algorithm would be used to solve this permutation problem and estimate the correct association of the diarization results among chunks. In order to classify speaker embedding easily, a scaled SPK loss was added to the total loss to optimize the model.

Before clustering the embedding vectors. A silent threshold would be applied to judge if a speaker is absent. For example, in Figure 1 there is only one active speaker in Chunk2. After getting rid of the silent speakers. Then those speaker embedding would be clustered according to the speaker identity to get the final diarization result.

2.2. Formulation of Neural Network

The Neural Network in Figure 1 can jointly estimate diarization results and speaker embedding. This NN can simply be formulated as:

$$\hat{Y}_i, \hat{E}_i = NN(X_i) \tag{1}$$

Where $\hat{Y}_i = \hat{y_{t,i}} \mid t = 1, ..., T \in R^{S_{Local} \times T}$ is the estimation of diarization results of the Chunk i that corresponds to X_i and $\hat{E}_i = (e_{i,s} \mid s = 1, ..., S_{Local}) \in R^{S_{Local} \times C}$ is the corresponding C-dimensional speaker embeddings.

2.3. Dataset

EEND and its variants are efficient in speaker diarizaton task. However all those models still require large amounts of annotated data for training. But the fact is the shortage of annotated speech data.

To solve this problem, a mixture simulation method was proposed[4]. It can be referred to as the concat-and-sum approach. In order to create a N-speaker mixture, it randomly chooses N sets from the whole speaker and concatenates those utterances with a silence. After that background noise and impulse noise would be added to increase the diversity of dataset.

3. Proposed Method

We describe our proposed modifications in the following subsections.

3.1. Data Simulation

We explore methods that can simulate data in a natural way. Inspired by [11,12], we first generated a set of simulated mixtures as Pre-training set and then generated a simulated conversations set as Fine-tune set.

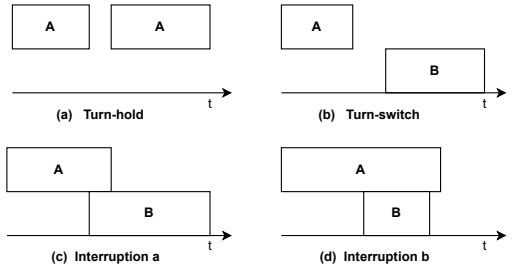

Figure 2: *Utterance transition cases.*

3.1.1. simulated mixtures

Based on [11], we analyzed the Magicdata-RAMC dataset and arranged the utterances as in a real conversation. Instead of using four transition types between utterances(TH, TS, IR, BC). We merged BC into IR as a subset (e.g., Interruption a, b), shown as Figure 2. One reason is that our dataset merely has BC situations. Besides that, we assume that if the duration of interruption were short enough, BC would become a subset of IR. For the TH and TH we assume the pause δ follows a exponential distribution, defined as:

$$\delta_{TH,TS} = exp(\beta_{TH,TS}) \tag{2}$$

Where β is the average pause value we calculate from the statistics. As for the IR we use as uniform distribution to simulate the overlap, as:

$$\delta_{IR} = \frac{1}{half\ duration\ of previous\ utt} \tag{3}$$

With the short utterances we split from Magicdata-RAMC we generated simulated mixtures as pre-training training and validation sets.

3.1.2. simulated conversations

As for the fine-tune set, we borrowed the method proposed in [12]. This paper proposes an alternative method for creating synthetic conversations that resemble real ones by using statistics about distributions of pauses and overlaps calculated from real conversation data. The statistics include:

- Distribution of pause in Turn-hold situation. Defined as $D_{=speaker}$.
- Distribution of pause in Turn-switch situation. Defined as $D_{\neq speaker}$.
- Distribution of overlap in Interruption situation. Defined as $D_{overlap}$.
- Probability of having pause in between two utterance of different speakers.

With the statistics calculated from the dataset we want to simulate, we can generate a dataset of simulated conversions. Using this method we significantly enlarged our fine-tuned dataset.

3.2. Conformer

Proposed as an alternative network architecture to Transformer[13], Conformer[14] has shown its superiority first at the ASR task. It has later been proven the effectiveness in dealing with sequential data such as audio. Compared to transformer it better captures the local frequency feature without losing the time domain global context information. To better utilize the benefit of local and global information, we replaced the Transformer encoder with the Conformer encoder. A typical Conformer encoder usually composed of several same Conformer block. And the Conformer block is composed of four modules stacked together, including an FFN module, a self-attention module, a convolution module, and a second FFN module in the end. Given the input E_i to the i-th Conformer block and E_{i+1} as the output. A Conformer block can simply be formulated as following:

$$\tilde{E}_i = E_i + \frac{1}{2}FFN(E_i) \tag{4}$$

$$\tilde{E}_i^{'} = MHSA(\tilde{E}_i) \tag{5}$$

$$\tilde{E}_i^{''} = Convolution(\tilde{E}_i^{'}) \tag{6}$$

$$E_{i+1} = LayerNorm(\tilde{E}_i^{''} + \frac{1}{2}FFN(\tilde{E}_i^{''})) \tag{7}$$

3.3. Additive Margin Penalty and Label Smoothing

Additive Margin Loss[15] was first proposed to solve the face verification problem. In classification, a decision boundary will be created to separate the classes. However, this can be a problem when the data of different classes lie near to the boundary. AM was designed to reduce the intra-class variance while increasing the separability of the class by introducing a margin penalty to the target class logit. Inspired by it, [16] modified and proposed an Additive Margin Penalty suitable to the BCE with a logit loss function.

After analyzing the PIT loss and SPK loss in our model, we simply exclude the scale factor to balance those two losses. The formulation as following:

$$\hat{\phi} = \delta(\varphi - m\phi + m(1-\phi)) \tag{8}$$

where $\phi \in 0, 1$ is the label, $\hat{\phi}$ is the posterior probability with additive margin, φ is the input logit and m is the additive margin value.

We also tried Label Smoothing[17] operation with the loss function. Which is viewed as a regularization technique. For BCE, we just simply replace label 0 with a penalty ϵ (e.g., 0.05) and label 1 with $1 - \epsilon$.

3.4. Spectral Clustering

As we mentioned before, clustering plays an important role in EEND-vector clustering diarization systems. Especially dealing with situations when the speaker number is unknown. In paper [10], several constrained clustering algorithms including COP-Kmeans, SC and Constrained AHC were evaluated on CALLHOME dataset, and Constrained AHC outperformed others. However, things change when the conversation becomes longer (e.g., greater than 30 min). Mainly because of the large number of embeddings and the manually controlled stopping criterion is hard to set up.

We investigated the Spectral Clustering method implemented in [3][18], and found its effectiveness on long audio speaker diarization problems. Thus we deploy it in our system. The SC algorithm consists of the following steps:

1. Calculation of the affinity matrix A, the cosine distance is applied for $A_{ij,(i\neq j)}$ and the maximal value for $A_{ii} = max_{j\neq i}A_{ij}$
2. Apply a sequence of refinement operations including Gaussian Blur, Row-wise Thresholding, Symmetrization, Diffusion and Row-wise Max Normalization.
3. Perform an eigen-decomposition on the refined affinity matrix.
4. Replace the ith segment embedding by corresponding top k eigen-value: $e_i = [v_{1i}, v_{2i}, ..., v_{ki}]$ and use a Kmeans++[19] to cluster these new embeddings and produce speaker labels.

3.5. Model Averaging

Model average as a way to ensemble multiple models, is widely used both in machine learning and deep learning research. In [4-10] a model average method was implemented to average the weights of models in a human defined specified range(e.g., epoch from 80 to 90). However, choose from consecutive range may not an optimized method. Because of the adjacent trained models might have a similar probability distribution.

In our system, after getting all the models. We first evaluate the models on the dev and test set to rank those models. Then based on the rank we simply choose the top K and do model average.

4. Experiments

In this section, we evaluate the effectiveness of our modified EEND-vector clustering diarization system in comparison

Table 1: *Training and Validation Datasets*

Dataset	Method	Duration
Pre-training Training Set	Simulated Mixtures	5261h
Pre-training Validation Set	Simulated Mixtures	86h
Fine-Tuning Training Set	Simulated Conversations	2480h
Fine-Tuning Validation Set	Simulated Conversations	48h

with the VBx[20] baseline[21], based on Magicdata-RAMC dataset[22].

4.1. Data

We summarize our datasets in Table 1. Consider of the shortage of real data, we use simulated data instead. For the pre-training datasets. We first segmented the MagicData-RAMC recordings into short utterances by the label files. A simulated mixture dataset is then created from the utterances using the method in section 3.1.1. We also include a few qualified parts of CN-Celeb[23] such as some sets labeled as speech. For the fine-tuning datasets, we also used the MagicData-RAMC segmented utterances, with the method in section 3.1.2. All the simulated mixtures and conversations only contained 2 speakers.

Both the training and validation set of pre-training and fine-tuning stage are generated from the same data source. And the essential parameters required by those two methods also computed on the MagicData-RAMC data set.

4.2. Experimental Settings

We basically followed the training procedure in [10]. We first trained the model on simulated 2-speaker mixtures for 100 epochs. Then, we fine-tuned the averaged model with the simulated 2-speaker conversation dataset for 50 epochs. We use a 6 layer Conformer encoder with 8 attention heads and 256 hidden units. The batch size is 64 and the chunk length is 50s. The default values of m in equation 8 is set to 0.25. We used the AdamW optimizer with 0.0003 as learning rate. For Conformer Block, the kernel size of Convolution layer is 11, and the number of hidden units in FFN module is 512. A spectral augmentation layer was added at the front of Conformer Block to randomly mask the input features in the time and frequency domains.

For inference, we set the threshold at 0.5 to determine speech activities and a median filter with 25 as kernel size is used to filter out short speech and break.

The evaluation metric we used is the Conversational Diarization Error Rate Metric(CDER), which can reasonably evaluate the performance of the speaker diarization system on the sentence level under conversational scenario.

4.3. Results

4.3.1. Experiments on Conformer

Table 2 shows the experiment result of introducing Conformer encoder to our system. As we can see, Conformer improves the CDER on all sets of data compared to the original transformer-based EEND-vector Clustering method. Thus, we continue our experiments with Conformer as our Encoder.

Table 2: *CDER(%) of Conformer and Transformer based EEND-vector clustering system and the VBx baseline. The third and forth column are CDER of Pre-training and Fine-tuned model respectively.*

Dataset	Encoder	Pretrain	Fine-tune
MagicData-RAMC Dev Set	Transformer	23.5	-
	Conformer	22.6	**21.9**
	VBx baseline	26.9	-
MagicData-RAMC Test Set	Transformer	25.3	-
	Conformer	24.8	**24.5**
	VBx baseline	28.2	-

4.3.2. Experiments on Additive Margin Penalty

We evaluated Additive Margin Penalty and Label Smoothing with our system and found that both these two techniques can accelerate convergence. However, compared to only with AM, applying them simultaneously can lead to a decline (24.8 to 25.4 on Test Set). This might be because both two operation work in a same way, but one on the logit and the other on the label.

4.3.3. Experiments on Sepctral Clustering

We evaluated AHC and SC algorithm based on MagicData-RAMC dataset. Though we know that all conversations contain 2 speakers. We still tested those two methods without this given information. For the constrained AHC, the cannot-link threshold κ was set at 10000, and the stopping criteria set at 1. As for the SC, we set minimum and maximum at 2 and 7 respectively. As we can see in Table 3. In both condition, the SC outperform the AHC method on MagicData-RAMC dataset.

Table 3: *CDER(%) of fine-tuned conformer model with AHC and SC algorithms for MagicData-RAMC dataset. The second column represent if the speaker number is given.*

Method	Speaker Num Info	Dev	Test
AHC	yes	27.9	31.4
SC	yes	24.3	**24.2**
AHC	no	32.3	32.7
SC	no	**21.9**	24.5

5. Conclusions

In this work, we improved the EEND-vector clustering system by modifying several techniques and we summarized them as following: First, we demonstrated that simulated training data naturally enhanced the original system. Furthermore, by fine-tune the averaged model on a simulated conversation dataset, an extra improvement is obtained. Second, by replacing the transformer encoder with a conformer encoder, the system is more capable if handling the uncertainty of data. We also tested some loss function techniques, and the results show that an additive margin penalty can accelerate model convergence and reduce intra-class variance, whereas label smoothing fails to do so. Finally, we demonstrated that SC outperforms AHC on long-form conversations, no matter whether the speaker number is given or not. Our system achieved our best performance on the MagicData-RAMC dev and test set with 21.9% and 24.5% CDER. And a second place in ISCSLP 2022 CSSD Challenge.

6. References

[1] G. Sell and D. Garcia-Romero, "Speaker diarization with PLDA i-vector scoring and unsupervised calibration." in *Spoken Language Technology Workshop (SLT), IEEE, 2014*, pp.413-417, 2014.

[2] D. Garcia-Romero, D. Snyder, G. Sell. D. Povey and A. McCree, "Speaker diarization using deep neural network embeddings." in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) IEEE, 2017*, pp.4930-4934, 2017.

[3] Q. Wang, C. Downey, L. Wan, P. A. Mansfield and I. L. Moreno "Speaker diarization with LSTM." in *in Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing(ICASSP)*, pp.5239-5243, 2018.

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

[5] Y. Fujita, N. Kanda, S. Horiguchi, Yawen. Xue, K. Nagamatsu and S. Watanabe "End-to-end neural speaker diarization with self-attention." in *IEEE ASRU 2019*, pp.296–303, 2019.

[6] S. Horiguchi, Y. Fujita, S. Watanabe, Yawen. Xue and K. Nagamatsu "End-to-end speaker diarization for an unknown number of speakers with encoder-decoder based attractors." in *Interspeech 2020*, pp.269–273, 2020.

[7] Y. Fujita, S. Watanabe, S. Horiguchi, Yawen. Xue, J. Shi and K. Nagamatsu "Neural speaker diarization with speaker-wise chain rule." in *arXiv preprint arXiv:2006.01796, 2020.*

[8] D. Yu, M. Kolbæk, Z.-H Tan and J. Jensen "Permutation invariant training of deep models for speaker-independent multi-talker speech separation." in *ICASSP 2017*, pp.241–245, 2017.

[9] K. Kinoshita, M. Delcroix, and N. Tawara "Integrating End-to-End neural and clustering-based diarization: Getting the best of both worlds." in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, 2021*, pp. 7198–7202, 2021.

[10] K. Kinoshita, M. Delcroix, and N. Tawara "Advances in integration of end-to-end neural and clustering based diarization for real conversational speech." in *Interspeech 2021*, pp. 3565–3569, 2021.

[11] N. Yamashita, S. Horiguchi, and T. Homma "Improving the Naturalness of Simulated Conversations for End-to-End Neural Diarization." in *arXiv preprint*, arXiv:2204.11232, 2022.

[12] F. Landini, A. Lozano-Diez, M. Diez and L. Burget "From Simulated Mixtures to Simulated Conversations as Training Data for End-to-End Neural Diarization." in *arXiv preprint*, arXiv:2204.00890, 2022.

[13] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N.Gomez, L. Kaiser, and I. Polosukhin "Attention is all you need." in *Advances in neural information processing systems*, pp. 5998–6008, 2017.

[14] A. Gulati, J. Qin, C.C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu, and R. Pang "Conformer: Convolution-augmented Transformer for Speech Recognition." in *Proceedings Interspeech 2020*, pp. 5036–5040., 2020.

[15] F. Wang, J. Cheng, W. Liu and H. Liu "Additive margin softmax for face verification." in *IEEE Signal Processing Letters 2018*, vol.25, no.7, pp.926–930, 2018.

[16] TY. Leung, and L. Samarakoon "Robust End-to-End Speaker Diarization with Conformer and Additive Margin Penalty." in *Interspeech 2021*, pp. 3575-3579, 2021.

[17] R. Müller, S. Kornblith and G. E. Hinton "When does label smoothing help?" in *Advances in Neural Information Processing Systems 2019*, pp. 4696-4705, 2019.

[18] W. Xia, H. Lu, Q. Wang and et al., "Turn-to-diarize: Online speaker diarization constrained by transformer transducer speaker turn detection." in *IEEE International Conference on Acoustics, Speech and Signal Processing 2022*, pp. 8077-8081, 2021.

[19] D. Arthur and S. Vassilvitskii, "k-means++: The advantages of careful seeding," in *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pp. 1027–1035, 2007.

[20] F. Landini, J. Profant, M. Diez, L. Burget, "Bayesian HMM clustering of x-vector sequences (VBx) in speaker diarization: theory, implementation and analysis on standard tasks." in *Computer Speech and Language 2022*, vol.71, p.101254, 2022.

[21] G. Cheng, Y. Chen, R. Yang, Q. Li and et al, "The Conversational Short-phrase Speaker Diarization (CSSD) Task: Dataset, Evaluation Metric and Baselines." in *arXiv preprint*, arXiv:2208.08042, 2022.

[22] Z. Yang, Y. Chen, L. Luo, "Open Source MagicData-RAMC: A Rich Annotated Mandarin Conversational(RAMC) Speech Dataset." in *arXiv preprint*, arXiv:2203.16844, 2022.

[23] Fan, Yue and Kang, JW and Li, LT and Li, KC and Chen, HL and Cheng, ST and Zhang, PY and Zhou, ZY and Cai, YQ and Wang, Dong. "CN-CELEB: a challenging Chinese speaker recognition dataset." in *ICASSP 2020* , pp.7604-7608, 2020.