On the Use of Speaker Information for Automatic Speech Recognition in Speaker-imbalanced Corpora

Kak Soky*†, Sheng Li†, Masato Mimura*, Chenhui Chu* and Tatsuya Kawahara*
* Graduate School of Informatics, Kyoto University, Kyoto, Japan
† National Institute of Information and Communications Technology (NICT), Kyoto, Japan
E-mail: {soky,mimura,kawahara}@sap.ist.i.kyoto-u.ac.jp

***Abstract*—We address the effective use of speaker information for automatic speech recognition (ASR) in a speaker-imbalanced dataset. Recently, joint speaker and speech recognition has been investigated in end-to-end (E2E) systems. However, speaker information as the output of speaker recognition (SRE) is not explicitly used for ASR in these systems. Inspired by speaker embedding for ASR, we propose a direct connection of SRE to the ASR decoder. The E2E architecture allows for backpropagating the ASR loss to the SRE decoder, resulting in joint optimisation. The architecture is beneficial for speaker-sparse datasets such as meetings and low-resource language settings, in which speaker clustering is conducted to compensate minor speakers. We also make a systematic comparison of our proposed method with other methods, including multi-task learning (MTL), adversarial learning (AL), and speaker attribute augmentation (SAug). It is shown that the use of speaker cluster information improves both ASR and SRE, and the proposed method outperforms other methods. It reduces errors of the baseline model by 3.35% and 8.23% for ASR and SRE, respectively.**

I. Introduction

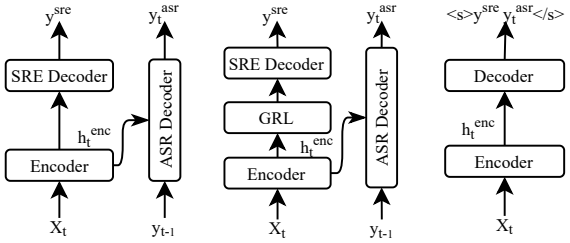
Speech technology has been significantly improved in the last decade with the advancement of deep learning techniques and computing resources. With this advancement, end-to-end (E2E) modelling [1], [2], [3], [4], [5] solves the complex problem of sequence labelling between input speech and output labels. It has been applied to automatic speech recognition (ASR) and speaker recognition (SRE), achieving promising results. ASR and SRE are complementary to each other. This means that we can decipher speech content and other meta information together and simultaneously. In other words, when we identify speakers, it is often easy to recognise their speech.

In this context, several previous studies investigated the embedding of speaker information into E2E ASR systems. In [6], speaker-representing features were extracted using a sequence summary network and then added to the encoder input as auxiliary features. Instead of using i-vectors directly as speaker embedding, Fan et al. [7] generated speaker embedding by concatenating the attention of the encoder output to i-vectors at each time step. Similarly, a speaker-aware persistent memory [8] concatenated i-vectors to the speech encoder self-attention part of the Transformer [5]. Within the same architecture, Shetty et al. [9] studied the effectiveness of providing speaker information on ASR, such as one-hot speaker vector and x-vector embedded into input and output of the encoder, and Sari et al. [10] proposed the speaker embedding by concatenating the memory vector (M-vector), a memory block that holds the extracted speaker i-vectors from training data and relevant i-vectors from the memory through an attention mechanism, to the acoustic features or to the hidden layer. In this approach, although speaker information is used to improve ASR, it neither explicitly conduct SRE nor use the supervision of the speaker information for ASR.

Another approach is joint SRE and ASR. Multi-task learning (MTL) is introduced to unify the training of transcribing the speech and identifying the speakers simultaneously by sharing the same speech feature extraction layers [11], [12], [13]. Adversarial learning (AL) adopts a similar architecture to that of MTL but learns a speaker-invariant model so that it is more generalised to new speakers by reducing the effects of speaker variability [14], [15], [16], [17], [18]. Most recently, speech attribute augmentation (SAug) was introduced as a fully E2E system integrating SRE and ASR; SAug embeds the speaker attribute tags into the training label and generates those tags together with the transcription in a single encoder-decoder model [19], [20], [21]. Unlike the speaker embedding approach, however, these methods do not use the speaker information explicitly for ASR.

Generally, E2E models require a large amount of speech corpus and work well with a balanced amount of speech per speaker, as in the cases of Librispeech [22], TEDLIUM [23] and the Corpus of Spontaneous Japanese [24]. On the other hand, in many low-resource languages, this assumption does not hold and utterance amounts over speakers are unbalanced, in that there are often dominant speakers and auxiliary speakers. This is called the class imbalance or speaker imbalance problem. It also occurs in resource-rich languages in many cases such as TV programs, meetings, and court proceedings, in which there is a limited set of speakers.

In this work, we address the effective use of speaker information for ASR and also tackle the speaker-imbalanced problem using the corpus of the Extraordinary Chambers in the Courts of Cambodia (ECCC). We identify major speakers and compensate minor speakers by clustering them. Inspired by the speaker embedding for ASR, we propose an extension of MTL that shares the encoder for SRE and ASR, and takes the speaker output of SRE as the speaker embedding, then feeds it to the ASR decoder. We investigate the effectiveness of using this speaker embedding in the Transformer decoder. We also compare our proposed method with MTL, AL, and SAug systems, which perform the SRE and ASR simultaneously. We



(a) MTL (b) AL (c) SAug

Fig. 1: Overview of joint speaker and speech recognition methods. MTL: multi-task learning, AL: adversarial learning, SAug: speech attribute augmentation.

find that our proposed method improves the performance of both SRE and ASR concurrently.

The rest of this paper is structured as follows. Section II gives an overview of joint speaker and speech recognition methods. We present the detailed concept of our proposed method in Section III. In Section IV, we describe the setup of the experiments and the results. We conclude the paper in Section V.

II. Joint Speaker and Speech Recognition

In this section, we review previous methods for joint speaker and speech recognition. We present the system architectures built on top of the Transformer architecture to produce the speaker ID and speech transcription in single or multiple decoders.

A. Multi-Task Learning (MTL)

In MTL, both SRE and ASR are given the same sequence of acoustic features $X_t = \{x_1, ..., x_n\}$ as input. A speaker ID, s, is predicted in SRE, whereas a sequence of vocabulary tokens $Y_t = \{y_1, ..., y_m\}$ is predicted in the ASR decoder. In this system, we can benefit from sharing the same encoder and employ a dual decoder of these tasks as shown in Figure 1a. The encoder and ASR decoder are based on the Transformer architecture, whereas the SRE decoder comprises two linear layers followed by the ReLU and softmax activation functions. Thus, in training this MTL network, we jointly optimise both SRE and ASR losses. The loss is therefore defined as:

$$\mathcal{L}_{total} = (1-\alpha)\mathcal{L}_{asr} + \alpha * \mathcal{L}_{sre}, \tag{1}$$

where α is the weight of the SRE task.

This joint recognition is possible when the number of speakers is limited and there is a large amount of data for each speaker. However, this method cannot be applied to the case of many speakers with little data for each, and thus we introduce clustering of the speakers.

B. Adversarial Learning (AL)

Similar but different from MTL, AL learns an acoustic representation that is speaker invariant to reduce the speaker variability by incorporating the adversarial loss of SRE, which is combined with the loss of ASR. This network has a similar architecture to MTL, but it uses the gradient reversal layer (GRL) shown in Figure 1b, which reverses the gradient of backward propagation [14].

Let the parameters θ_{enc}, θ_{asr} and θ_{sre} respectively denote the encoder, ASR and SRE decoders. The parameters are updated via back-propagation as follows:

$$\theta_{asr} \longleftarrow \theta_{asr} - \epsilon\frac{\partial\mathcal{L}_{asr}}{\partial\theta_{asr}}, \tag{2}$$

$$\theta_{sre} \longleftarrow \theta_{sre} - \epsilon\frac{\partial\mathcal{L}_{sre}}{\partial\theta_{sre}}, \tag{3}$$

$$\theta_{enc} \longleftarrow \theta_{enc} - \epsilon(\frac{\partial\mathcal{L}_{asr}}{\partial\theta_{enc}} - \lambda\frac{\partial\mathcal{L}_{sre}}{\partial\theta_{enc}}), \tag{4}$$

where ϵ is a learning rate and a negative coefficient -λ is used to remove the speaker variability from the speaker classification.

AL learns to improve the ASR as a main task, whereas SRE is an auxiliary task. AL intends to be robust for unseen speakers, but it does not leverage speaker information for ASR.

C. Speech Attribute Augmentation (SAug)

SAug is a fully E2E method integrating SRE and ASR in a single encoder-decoder architecture. The speech attribute is analogous to a language ID in a multilingual system. It can be a speaker ID, gender or age label [20]. Speech attributes are placed in front of the lexical token sequence of each utterance. Given a sequence of acoustic features X_t, a model must produce a label sequence $Y_t = \{s, y_1, ..., y_m\}$, where s is a speech attribute and y is a sequence of vocabulary tokens.

This network is usually trained to output the attribute label at the beginning of speech transcription for each utterance with a single decoder as shown in Figure 1c, and thus we do not have to prepare classifiers for the attributes explicitly. However, it is reported that the speaker ID attribute is not effective for improving ASR [20] because SRE and ASR are usually correlated negatively to each other. It is therefore impractical to improve the performance of these tasks together in a single decoder.

III. Proposed Method

As presented in Section II, MTL and AL do not use speaker information explicitly for ASR, whereas SAug uses a single decoder. In this study, we propose the direct use of the speaker embedding of the SRE output to the ASR decoder. Unlike the previous speaker embedding, the proposed architecture is an E2E network, conducting both ASR and SRE with supervision for speaker IDs. The proposed system is expected to be useful for speaker-sparse and imbalanced datasets. The speaker information is effective for major speakers, and speaker clustering is conducted for minor speakers. The proposed network injects a speaker output (y^{sre}) into the ASR decoder as shown in Figure 2. We investigate five options, namely self-attention (A), after self-attention (B), cross-attention (C), after cross-attention (D), and after the feed-forward network (E). Each of these methods is tested one by one, and the combination of two methods, such as AC and BD, is also evaluated. In this

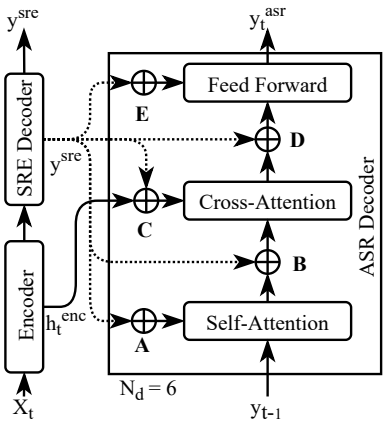


Fig. 2: Proposed method. $\oplus$ denotes the "sum" operation.

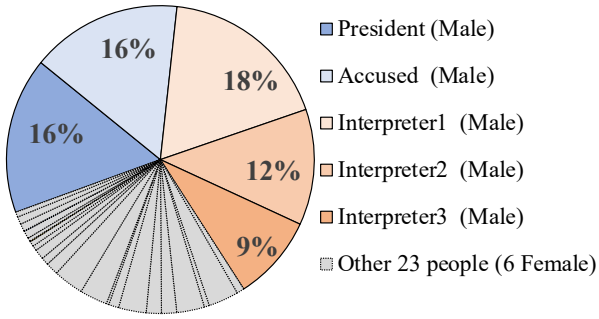


Fig. 3: Speaker distribution in the ECCC; five major speakers are the president of the chamber, the accused and three interpreters.

process, the ASR loss is backpropagated to the SRE network, which means SRE is enhanced based on ASR.

Let h_t^{enc}, y^{sre}, y_t respectively denote the encoder output, the SRE output, and the decoder input at time step t. The embedding operation comprises the weighted sum of h_t^{enc} and y^{sre} or that of y_t and y^{sre}. Note that the operation in B, D or E is merging between a residual network at the early time step $t-1$ (y_{t-1}) and the speaker information (y^{sre}). Meanwhile, in A, y^{sre} is merged with the key of self-attention at the previous time (K_{t-1}), and in C, y^{sre} is combined with the key of cross-attention at the current time (K_t).

IV. Experiments

A. Data setup

The ECCC is a court established to prosecute senior leaders who committed crimes during the period of Democratic Kampuchea, namely the Khmer Rouge regime in Cambodia from 1975 to 1979. We collected recordings of 222 sessions of the first caseload that spanned from February 17, 2009 to November 27, 2009. Each session had a length from 5 to 150 minutes and involved a wide range of speakers, including the indicted person, witnesses, judges, clerks, co-prosecutors, lawyers, civil parties and interpreters. The videos are uploaded to the Youtube[1], and the proceedings are published in a digital format at the ECCC official website[2].

We first built a large spontaneous speech corpus that is comprised of 78,903 utterances (about 186 hours) with 28 speakers (22 male and 6 female). We then randomly split this dataset into a test set and a training set by 5% and 95%, respectively as presented in Table I.

Figure 3 illustrates the speaker distribution in the ECCC corpus based on the duration of speech. It presents the measurement in percentage of each speaker. This pie chart shows that the dataset has a crucial speaker imbalance, in that five major speakers, the president of the chamber, the accused, and three interpreters, talk more than 70% of speech among all speakers. Similarly, it also shows that male speakers are dominant compared to female speakers that have a small proportion in the "Other people" group. It means that the gender-based classification is impractical. We thus classify the speakers into a group of six (Gr6), comprising the five major speakers and a combination of other speakers. Thus, we experiment on the original dataset of 28 speakers (Gr28) and Gr6 for our proposed method.

TABLE I: Data statistics used in this work

Dataset	#utterance (#hour)	#character
Training	75,170 (176)	6.02 M
Test	3,733 (10)	294 K
Total	78,903 (186)	6.30 M

B. System Configurations

1) Baseline System: We adopt a Transformer-based ASR system, which is comprised of the encoder block ($N_e = 6$) and the decoder block ($N_d = 6$) with the feed-forward inner dimension of 1024, the model dimension of 256, and the attention head number of 4, which are unchanged in all experiments. The 80-dimensional log-Mel filter bank features, which were mean and variance normalised per speaker, were extracted with a 10-ms frame shift of a 25-ms window. We then subsampled the input features using a two-layer time-axis CNN with ReLU activation with 256 channels, stride size 2, and kernel size 3. The model was jointly trained with CTC (weight $\alpha = 0.2$). The "noam" optimiser was used with 25,000 warmup steps and an initial learning rate of 5. The model was trained with ESPnet toolkit [25] using 32 batch size for 30 epochs on a 12-GB Titan X GPU.

For the baseline SRE, we separately experimented using the x-vector [26] following Kaldi's SRE16 recipe.

2) MTL and AL Systems: The MTL and AL network takes the 80-dimensional log-Mel filterbank features to produce a sequence of vocabulary tokens Y_t and a speaker label s separately. Here Y_t has 73 characters and s has 6 speaker IDs. The ASR decoder is the same as the baseline system, whereas

[1]www.youtube.com/user/krtribunal/
[2]www.eccc.gov.kh/

the SRE decoder takes a mean value of h_t^{enc} of the encoder output in 256 dimensions and feeds it to a linear layer followed by the ReLU activation function and then down-projects to six as the number of speakers using another linear layer. Finally, we use a softmax layer to generate the speaker label output. In the preliminary experiments, we tested $\alpha \in \{0.2, 0.5, 0.7\}$, and found that $\alpha = 0.5$ gives the best performance.

In the GRL of AL shown in Figure 1b, we multiply the gradient by $\lambda = -1$ to compute the reversed gradient at the backward propagation phase, whereas in the forward propagation phase, MTL and AL are acted in the same operation.

3) SAug System: The SAug system is a single encoder-decoder similar to the original Transformer architecture. The configuration of this model is therefore the same as that of the baseline system except for the output label. The six-speaker IDs were embedded as the speech attribute labels to ground-truth in training similar to [20]. These IDs are generated together with speech transcription. We thus calculate SRE performance with the speaker attribute label and simply remove this beginning attribute from the transcription and then calculate the character error rate (CER) for ASR performance.

4) Proposed System: We conduct experiments on both Gr28 and Gr6. For the speaker embedding operation, we investigate each option from A to E and layer-wise from lower to deeper by a single layer or multiple layers.

The summation of vectors is used in this operation. We take the speaker output vectors having 28 (Gr28) or 6 (Gr6) dimensions according to the number of speakers in the training set, which is enlarged via a linear layer to 256 dimensions to match the encoder layer output or decoder input. We then normalise this output using a layer normalisation [27] before executing the summation operation. Only in the C option, we sum with the output of the encoder. Otherwise, we sum with a residual output of the decoder module.

C. Results and Discussions

We evaluate the performance of all ASR models on the basis of the CER, whereas the SRE performance is on the basis of the ratio of utterances of incorrect prediction. Table II presents the best performance of ASR and SRE with each method. Only for the baseline system, ASR and SRE were conducted with different models, in which SRE is conducted with the x-vector model. The table shows that the SAug has a better result for SRE, but it is not as effective as MTL, AL and the proposed method in terms of ASR performance. This suggests that it is difficult to train the model in a single decoder. The use of AL improved the ASR, however, it does not work as SRE. MTL is effective for both tasks, but our proposed method is more effective than the other compared methods in the clustering (Gr6) settings. This demonstrates that embedding speaker information to the ASR decoder does not only improve the ASR but also tune the performance of SRE. Moreover, Gr6 gives better performance than Gr28, showing that the combination of minor speakers is critical to solve the speaker-imbalanced problem.

Regarding our proposed method, we compared different options of the speaker embedding applied to all layers of the decoder. Table III shows that combined options AC is the most effective in both tasks. With this AC option, we tested the layer-wise performance by embedding the speaker information into a single layer or multiple layers. Table IV shows that embedding the speaker information into only a single layer is as effective as embedding into all layers in terms of ASR performance but slightly degrades the SRE performance. Moreover, it is shown that embedding speaker information into lower layers of the decoder shows better improvement for SRE and ASR together. This is reasonable as the speaker information is usually reduced in the ASR decoder.

In summary, our proposed method improved not only ASR but also SRE performance from the baseline model by 3.35% and 8.23% for ASR and SRE, respectively.

TABLE II: Comparison of all systems for SRE and ASR.

System	SRE (%incorrect)	ASR (%CER)
Baseline (Gr6)		
- X-vector	9.72	/
- Transformer	/	7.46
Joint speaker and speech recognition methods		
- MTL (Gr6)	9.09	7.30
- AL (Gr6)	75.16	7.30
- SAug (Gr6)	**8.81**	7.37
Proposed method		
- Gr6 (option AC; all layers)	8.97	**7.21**
- Gr28 (option AC; all layers)	11.27	7.26

TABLE III: Comparison of embedding options applied to all layers in our proposed method (Gr6)

Embedded option (all layers)	SRE (%incorrect)	ASR (%CER)
Option A	9.08	7.30
Option B	9.10	7.33
Option C	9.21	7.26
Option D	9.02	7.33
Option E	9.18	7.26
Option AC	8.97	**7.21**
Option BD	**8.92**	7.40

TABLE IV: Comparison of layer-wise applications of AC option of our proposed method (Gr6)

Embedded Layer (AC)	SRE (%incorrect)	ASR (%CER)
Layer 1	9.18	**7.20**
Layer 2	9.10	7.26
Layer 3	9.35	7.33
Layer 4	9.26	7.30
Layer 5	9.24	7.35
Layer 6	9.91	7.26
Layer 1,2	**9.02**	7.28
Layer 1,2,3	9.10	7.24
Layer 1,2,3,4	9.08	7.27
Layer 5,6	9.48	7.29
Layer 4,5,6	9.61	7.28
Layer 3,4,5,6	9.30	7.27

V. Conclusions

In this work, we present a method that integrates the speaker information into the ASR decoder and also address the problem of speaker-imbalanced problem in ASR by identifying major speakers and clustering other minor speakers. The proposed method outperformed MTL and AL in both ASR and SRE, and it outperformed SAug in term of ASR performance in a large margin. It has the potential to be extended to multilingual systems in the future.

Acknowledgment

We would like to thank the National Institute of Posts, Telecoms, and Information Communication Technology (NIPTICT), Phnom Penh, Cambodia for giving us the resources of ECCC corpus.

References

[1] A. Graves, S. Fernandez, F. Gomez, and J. Shmidhuber, "Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks," in *Proceedings of the 23rd International Conference on Machine Learning*, 2006.

[2] A. Graves, "Sequence Transduction with Recurrent Neural Networks," *ICML Representation Learning Workshop*, Nov. 2012.

[3] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 4960–4964.

[4] T. Hori, S. Watanabe, Y. Zhang, and W. Chan, "Advances in Joint CTC-Attention Based End-to-End Speech Recognition with a Deep CNN Encoder and RNN-LM," in *Proc. Interspeech 2017*, 2017, pp. 949–953.

[5] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. Gomez, L. Kaiser, and I. Polosukhin, "Attention is All You Need," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017.

[6] M. Delcroix and S. Watanabe and A. Ogawa and S. Karita and T. Nakatani, "Auxiliary Feature Based Adaptation of End-to-End ASR Systems," *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, vol. 2018-September, pp. 2444–2448, 2018.

[7] Z. Fan, J. Li, S. Zhou, and B. Xu, "Speaker-aware speech-transformer," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2019, pp. 222–229.

[8] Y. Zhao, C. Ni, C.-C. Leung, S. Joty, E. S. Chng, and B. Ma, "Speech Transformer with Speaker Aware Persistent Memory," in *Proc. Interspeech 2020*, 2020, pp. 1261–1265.

[9] V. M. Shetty, M. S. Mary N. J, and S. Umesh, "Investigation of Speaker-adaptation methods in Transformer based ASR," *CoRR*, vol. abs/2008.03247, 2020.

[10] L. Sari, N. Moritz, T. Hori, and J. Le Roux, "Unsupervised Speaker Adaptation Using Attention-Based Speaker Memory For End-To-End ASR," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. IEEE, Apr. 2020, pp. 7384–7388.

[11] Z. Tang, L. Li, and D. Wang, "Multi-task Recurrent Model for Speech and Speaker Recognition," in *2016 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*, 2016, pp. 1–4.

[12] N. Kanda, Y. Gaur, X. Wang, Z. Meng, Z. Chen, T. Zhou, and T. Yoshioka, "Joint Speaker Counting, Speech Recognition, and Speaker Identification for Overlapped Speech of Any Number of Speakers," in *Interspeech 2020*. ISCA, October 2020.

[13] N. Kanda, X. Chang, Y. Gaur, X. Wang, Z. Meng, Z. Chen, and T. Yoshioka, "Investigation of End-To-End Speaker-Attributed ASR for Continuous Multi-Talker Recordings," in *SLT 2021*. IEEE, January 2021.

[14] Y. Ganin and V. Lempitsky, "Unsupervised Domain Adaptation by Backpropagation," in *Proceedings of the 32nd International Conference on Machine Learning*, 07–09 Jul 2015, pp. 1180–1189.

[15] Y. Shinohara, "Adversarial Multi-Task Learning of Deep Neural Networks for Robust Speech Recognition," in *Interspeech 2016*, 2016.

[16] T. Tsuchiya, N. Tawara, T. Ogawa, and T. Kobayashi, "Speaker Invariant Feature Extraction for Zero-Resource Languages with Adversarial Learning," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 2381–2385.

[17] Z. Meng, J. Li, Z. Chen, Y. Zhao, V. Mazalov, Y. Gong, and B. Juang, "Speaker-Invariant Training Via Adversarial Learning," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 5969–5973.

[18] Z. Meng, J. Li, and Y. Gong, "Adversarial speaker adaptation," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 5721–5725.

[19] L. El Shafey, H. Soltau, and I. Shafran, "Joint Speech Recognition and Speaker Diarization via Sequence Transduction," in *Proc. Interspeech 2019*, 2019, pp. 396–400.

[20] S. Li, D. Raj, X. Lu, P. Shen, T. Kawahara, and H. Kawai, "Improving Transformer-Based Speech Recognition Systems with Compressed Structure and Speech Attributes Augmentation," in *Proc. Interspeech 2019*, 2019.

[21] H. Henry Mao, S. Li, J. McAuley, and G. W. Cottrell, "Speech Recognition and Multi-Speaker Diarization of Long Conversations," in *Proc. Interspeech 2020*, 2020, pp. 691–695.

[22] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 5206–5210.

[23] A. Rousseau, P. Deléglise, and Y. Estève, "TED-LIUM: an automatic speech recognition dedicated corpus," in *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*. Istanbul, Turkey: European Language Resources Association (ELRA), May 2012, pp. 125–129.

[24] K. Maekawa, H. Koiso, S. Furui, and H. Isahara, "Spontaneous Speech Corpus of Japanese," in *Proceedings of the Second International Conference on Language Resources and Evaluation (LREC'00)*, May 2000.

[25] S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N. Soplin, J. Heymann, M. Wiesner, N. Chen, A. Renduchintala, and T. Ochiai, "ESPnet: End-to-End Speech Processing Toolkit," in *Proc. Interspeech 2018*, 2018, pp. 2207–2211.

[26] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018.

[27] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer Normalization," *arXiv preprint arXiv:1607.06450*, 2016.