An Attention-based Neural Network on Multiple Speaker Diarization

Shao Wen Cheng
National Yang Ming Chiao Tung University
Hsinchu, Taiwan
shaowen.eic09g@nctu.edu.tw

Kai Jyun Hung
National Yang Ming Chiao Tung University
Hsinchu, Taiwan
erichong860711@gmail.com

Hsie Chia Chang
National Yang Ming Chiao Tung University
Hsinchu, Taiwan
hcchang@mail.nctu.edu.tw

Yen Chin Liao
National Yang Ming Chiao Tung University
Hsinchu, Taiwan
ycliao.ee92g@g2.nctu.edu.tw

Abstract-Speaker diarization is a task to label audio or video recordings with classes that correspond to speaker identity for each point in time, which can be used in a multi-speaker conversation environment, such as a meeting or interview. Moreover, speaker diarization can be used to improve the performance of auto speech recognition. This paper presents an end-to-end diarization model based on an attention mechanism with data augmentation, several data pre-processing, and post-processing. In the CALLHOME data set, the case of two speakers reached a 9.12% diarization error rate.

We combine the speaker diarization model, and auto speech recognition model and implement the transcript conversion system on an edge device. By using proposed speaker diarization as preprocessing to segment recording according to different speakers, then get the transcript of each segmented utterance by ASR model to fulfill the transcript conversion on the edge device. Experiment shows that our model also performs well on edge devices with both accuracy and inference time.

Keywords—Speaker Diarization, End-to-end Diarization Model, Attention Mechanism, Transcript Conversion

I. INTRODUCTION

Auto speech recognition (ASR) becomes possible due to the development of machine learning and deep learning. However, ASR is not robust enough, for example, it may not perform well due to overlapping speech in multi-speaker conversation scenarios. How to segment recordings based on different speakers is a key task and that is what the diarization model trying to deal with. The traditional diarization framework splits diarization into multiple tasks, voice activity detection (VAD), feature extraction, and clustering. It is time-consuming because those models should be trained separately. Moreover, the diarization system may not perform well in some application scenarios when running on an edge device with lower computing performance. Therefore, the end-to-end speaker diarization system came out to give a different view. Different from speaker recognition, speaker diarization needs to divide the audio or video recordings based on different speakers which means there are multiple speakers in one recording.

At the earliest, speaker diarization is mostly composed of VAD, embedding (feature extraction), and clustering. Most papers consider VAD as a separate task and usually ignore this part. Moreover, most of the papers ignored the overlapping speech issue due to using the clustering method. In 2019, Hitachi published ” End to End Neural Diarization” (EEND) [1][2]. Different from previous works, EEND doesn’t have multiple separated models. It took the speaker diarization task as a multi-label classification problem, so it can deal with overlapping speech and the function of VAD is also included. In this paper, our model is based on an attention mechanism to achieve an end-to-end model with more effective data pre-processing, post-processing, and layer design to increase the performance on speaker diarization tasks.

The rest of this paper is organized as follows. Chapter 2 provides the essential information and knowledge of this thesis. Chapter 3 introduces the data preprocessing, postprocessing, data augmentation, and the proposed diarization model. Chapter 4 shows the experiment and implementation result. Conclusions are provided in chapter 5.

II. BACKGROUND

2.1 Attention Mechanism

In 2017, Google published ”Attention is all you need ” [3]. It was used for dealing with natural language processing at the beginning. Before the attention mechanism was published, natural language processing has been being the field of recurrent neural networks (RNN) or long short-term memory (LSTM). RNN will have a vanishing gradient problem if the input sequence is too long. Gradients convey information used in RNN parameter updates, and when the gradient shrinks, the parameter updates become minor, implying that no meaningful learning is taking place. LSTM is an improved version of RNN that avoid vanishing gradient problem, but it is hard to compute parallel and still not very sensitive to the very first input. Attention mechanisms can improve these problems.

Attention mechanism can be seen as mapping a query and a set of key-value pairs to an output. The query, keys, and values are vectors. The output is a vector computed as a weighted sum of the values, where the weights come from the query and corresponding key. The attention mechanism is allowed to compute parallel. Moreover, every input is equally important and sensitive which makes the attention mechanism has a better performance on NLP. After that, researchers start to adopt an attention mechanism to deal with other tasks, diarization is also one of them.

2.2 Transformer

Transformer was proposed in the mentioned [3]. It is a Sequence to Sequence architecture that contains two-part, the encoder and the decoder. Encoder consists of several encoder layers, so as the decoder. Encoder layers include multi-head attention layer, add & norm layer, and feedforward layer. Multi-head attention layer is similar to the attention mentioned above. Multi-head can focus on the different targets to improve the performance. Add & norm layer include a residual connection for preventing vanishing gradient and layer normalization.

Transformer is used as a Sequence to Sequence model at first. Later, people start to adopt the encoder or decoder individual. Bidirectional Encoder Representations from Transformers (BERT) [4] used only the encoder part but still get a good performance. That shows the potential and flexibility of the transformer.

2.3 Permutation-Free Objectives

In the diarization task, the model should label the input recordings with classes that correspond to different speakers for each point in time, so there have multiple speakers in one recording. In other words, speaker diarization needs to separate different speaker's speaking duration. Different from speaker recognition, diarization doesn't need to register the speaker's identity before inference, which means the ground truth has multiple combinations. If there are k speakers in the recording, the number of combinations is k!. Permutation-Free objectives [5][6] can describe this situation. This method makes the diarization task different from other deep learning tasks because the ground truth is nonuniqueness.

III. Proposed Method

3.1 Data Preprocessing

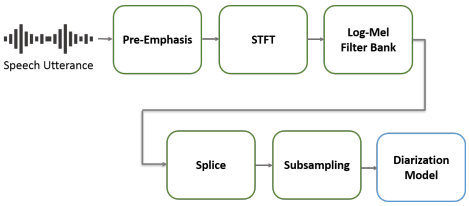

Fig. 1. Block diagram of data preprocessing

Pre-emphasis is implemented by subtracting each time's value from the previous time's value by a constant ratio α which is usually set as 0.97. Pre-emphasis increases the amplitude of high-frequency bands and decreases the amplitudes of lower bands. This method highlights the feature differences before and after which can make the signal difference more delicate for extracting useful features. The comparison between adopting pre-emphasis or not is shown in Fig. 2 and Table 1. Noted that the signal is focused on the difference between time and it also improves the signal-noise ratio.

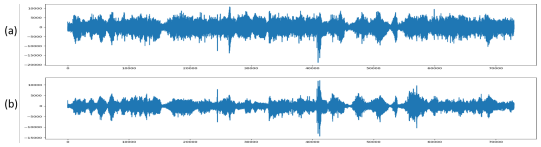

Fig. 2. Difference between adopting pre-emphasis (b) or not (a)

TABLE 1. COMPARISON BETWEEN USING PRE-EMPHASIS OR NOT

Method	Diarization Error Rate (DER)
Without pre-emphasis	9.77%
With pre-emphasis	9.57%

Short-term Fourier transform, log-Mel filter band, splicing, and subsampling are also implemented in the data preprocessing stage.

3.2 Data Augmentation

Each recording may have varying degrees of noise caused by the recording environment or equipment and the domain we try to apply transfer learning to is a lack of training data might decrease the performance of the Diarization result.

We use three data augmentation methods to artificially increase the size of a training set by modifying existing data, which is an effective method to avoid overfitting. The first technique is pitch shifting, which is a process of changing the pitch of input recording without affecting its speed. And the second one is noise adding, which involves the addition of white noise. It is critical because we have mentioned that our dataset may have varying degrees of noise and it can make the model more robust. The last one is SpecAugment [7] which was published by Google in 2019 to improve the performance of speech recognition. It tries to augment spectrogram directly which makes SpecAugment less time-consuming. SpecAugment includes three different methods, time warping, frequency masking, and time masking. Our work adopts frequency masking because the diarization task is sensitive along the time axis. Using time warping and time masking may shift or mask the speaking duration which makes the model have wrong or empty information to discriminate the speaker.

3.3 Data Postprocessing

Fig. 3. Block diagram of data postprocessing.

3.3.1 Utterance Reconstruction

Due to the characteristic of the attention mechanism, the length of the input sequence has an upper bound which means the transformer diarization system is difficult to deal with the long recording. In the training stage, we can directly split those data and the ground truth into pieces, but it is impracticable in the inference stage due to the hypothesis being permutation-free objectives, these segmented hypotheses can't be concatenated directly. Also, defining the identity of speakers by the order of who speaks is not effective here because the input is segmented.

To solve these problems, we segment the input recording into several utterances which are able to feed into the model individually. Each segmented utterance has an overlap part with the previous utterance. After inference, we will have the hypothesis of each utterance which showed in Fig. 4.

Fig. 4. Segment the input recording

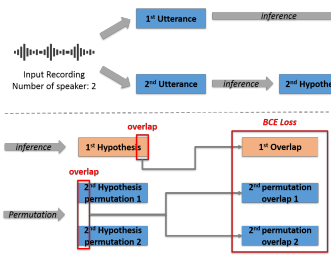

Fig. 5. Calculating BCE loss of the overlap part.

Fig. 5. shows how to use the overlap part of hypotheses to determine the combination of utterances. It shows a case that which the input recordings are split into two utterances, and the number of speakers is 2. The first utterance is directly fed into the model, and its hypothesis of it is a

criterion for the following utterances. After we get the hypothesis of the second utterance, we permute the hypothesis. Because the number of speakers is two, we will get two permuted hypotheses. Then, we calculate the binary cross-entropy loss of the overlap between the permuted hypotheses and the first hypothesis. The permuted hypothesis with the least loss is concatenated with the first hypothesis. If the input recording is split into k utterances, this procedure should be repeated k−1 times. The overlap factor in this paper is 100.

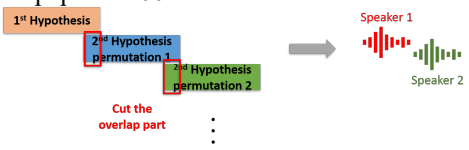

Fig. 6. Concatenate the split utterance.

3.3.2 *Threshold Choosing*

The output of the diarization system is a probability map, so we transfer the probability map to a binary map by using a threshold. The threshold affects performance a lot, so this is crucial to select the value. Fig. 7 shows the experiment result of the varying threshold.

We use binary search to adjust the threshold to be closer to the best value for finding the lowest point of DER. The binary search iteration will be executed in training stage and the DER of every epoch will be compared to the previous DER. If the DER is lower, the threshold will be recorded as the best threshold. If the best threshold remains unchanged for 10 epochs, the iteration will be stopped and set this value as the best threshold then adopted it in the inference stage.

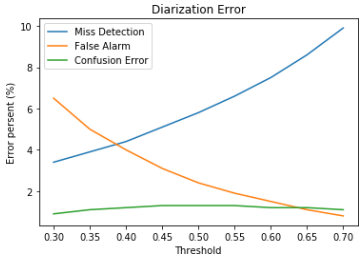

Fig. 7. DER of different thresholds.

3.4 *Proposed Diarization Model*

3.4.1 *Model Structure*

The model can be divided into three parts, encoder layer, transformer encoder, and decoder layer. The overview of the model is shown in Fig. 8.

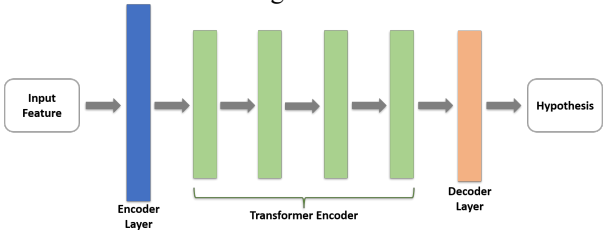

Fig. 8. Overview of the proposed model

The first part is the encoder layer which is used to reduce the input dimension from 345 to 256. We implement two different layers, linear layer, and 1D convolution which are shown in Fig. 9. The advantage of utilizing 1D convolution for sequence classification is that it can learn directly from raw time series data because the kernel slides along the time axis. To achieve dimension reduction, the output channels are 256, kernel size is 3, the stride is 1 and zero padding is used.

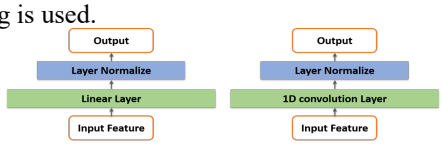

Fig. 9. Two kinds of Encoder Layer.

The second part is the transformer encoder which is composed of 4 transformer encoder layers. Each encoder layer is the same and shown in Fig. 10 with skip connection. The number of heads for the multi-head attention is 4.

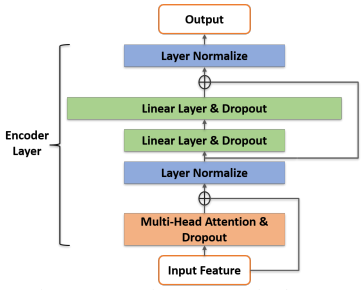

Fig. 10. Transformer encoder layer.

The third part is the decoder layer which is a simple linear layer and sigmoid function to turn previous processed features into a probability map. In this paper, the number of speakers is fixed as 2, so the output dimension is 2.

3.5 *Simulated Mixtures*

Due to lacking the training data, we need to create training data by ourselves. [1] and [8] show an algorithm to mixture speech recognition data into labeled diarization data, which are also called simulated mixtures.

There are two different sets of speakers used in this paper, one includes Switchboard- 2 (Phase I, II, III), Switchboard Cellular (Part 1, Part2), and NIST Speaker Recognition Evaluation corpora (2004, 2005, 2006, 2008). Another one is Mini LibriSpeech ASR corpus, which is a subset of LibriSpeech ASR corpus. The set of background noise is from MUSAN corpus. The set of room impulse response I is from the dataset used.

In this paper, each training utterance is composed of 10 to 20 utterances per speaker. The interval factor β is 2, 3, and 5. The simulated mixtures include training set and testing set, the training set is used in the training stage, and the testing set is used to evaluate performance in chapter 4.

3.6 *Training of Diarization System*

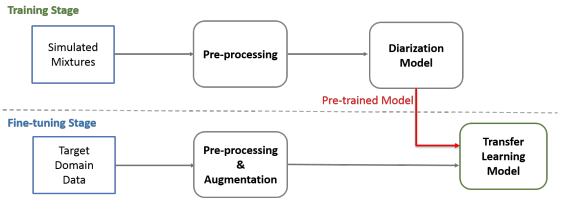

Fig. 11. Training flow of proposed model.

Fig. 11 shows the training flow of the proposed model. The loss function for the proposed diarization model is binary cross-entropy, which is usually used to deal with multi-label tasks. The diarization model is trained on simulated mixtures at first, then fine-tuning the pre-trained model on another real dataset with a small amount of data.

The input sequence of the training stage and fine-tuning stage is 500. This means the longest length of input recordings with a sample rate of 8000 is 50 seconds because we want to make the batch size to be larger to speed up the convergence of the model.

IV. Experimental Results

We use the evaluation metrics for the diarization task is diarization error rate (DER) which is the standard metric for evaluating and comparing speaker diarization systems which can be calculated as follows.

$$DER = \frac{MissedDetection + FalseAlarm + Confusion}{Reference}$$

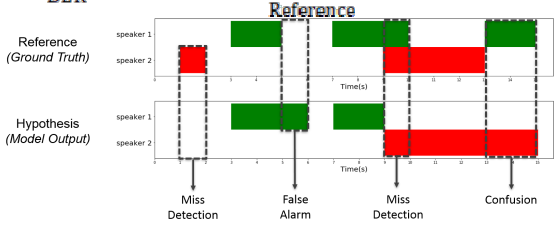

Fig. 12. Errors of diarization task

We evaluate the proposed model on real diarization data, called CALLHOME, and compare it with other works SA-EEND [9] and SA-EEND-EDA [10]. CALLHOME is disk 8 of NIST Speaker Recognition Evaluation which includes 2 to 7 speakers in each recording. To compare with SA-EEND and SA-EEND-EDA, we only adopt recordings with 2 speakers then split these recordings into fine-tune set and testing set, which include 155 and 148 recordings respectively. The recordings in the fine-tune set and testing set are totally identical to the com- pared works. Our proposed models, SA-EEND and SA-EEND-EDA use the same speaker set and noise set of simulated mixtures mentioned in section 3.5 as the training set. The training set of SA-EEND is the same as ours, but the training set which SA-EEND-EDA used is 4 times larger than ours. All the models use the same fine-tune set to fine-tune. Table II shows that both of our proposed models perform better than SA-EEND. However, we can find that performance of our proposed models is worse than SA-EEND-EDA. The reason is that the number of training sets is different. Moreover, SA-EEND-EDA is based on transformer encoder and LSTM, the model parameters and size are greater than ours. In other words, it can be speculated that our work can use less inference time to achieve nearly performance, or even obtain better performance if we use larger datasets with huge computation resources.

Moreover, we combine speaker Diarization, auto speech recognition (ASR), and transcript conversion and implement it on Nvidia Jetson AGX Xavier. By using proposed speaker diarization as preprocessing to segment recording according to different speakers, then get the transcript of each segmented utterance by ASR model to fulfill the transcript conversion on the edge device.

TABLE II. SPEAKER DIARIZATION ON CALLHOME

Model	DER
Proposed convolution model of 2 speakers	9.47%
Proposed linear model of 2 speakers	9.12%
SA-EEND [9]	9.54%
SA-EEND-EDA [10]	8.07%

V. Conclusion

In this paper, we successfully implement a speaker diarization application. In data preprocessing, pre-emphasis, short-term Fourier transform, and log Mel filter bank is used for making the model easier to extract the features of input data. During the model construction, we adopt two different architectures based on the transformer encoder. The data augmentation performs well if the training data is not enough. Also, data augmentation can prevent overfitting. In the postprocessing, utterance reconstruction makes the proposed model able to handle the long conversation. Median filter reduces the influence of noise. Threshold choosing allows us to find the proper threshold ratio to binary the model's hypothesis as a standard diarization result. In the training stage, we first train the model on simulated mixtures. Then in the fine-tuning stage, we fine-tune our model on the real corpora, which is CALLHOME training set. The proposed diarization model can achieve 9.12% of DER on CALLHOME testing set. Achieving better multi-people speaker diarization or even fulfilling online speaker diarization will be our goal in the future.

We also combine our proposed diarization model and auto speech recognition model and implement the transcript conversion system on an edge device. Experiment shows that our model also performs well in the scenario with two people on edge devices with both accuracy and inference time.

VI. References

[1] 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.

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

[3] A. Vaswani, N. M. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," ArXiv, vol. abs/1706.03762, 2017.

[4] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in NAACL-HLT, 2019.

[5] J. R. Hershey, Z. Chen, J. Le Roux, and S. Watanabe, "Deep clustering: Discriminative embeddings for segmentation and separation," in IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2016, pp. 31–35.

[6] D. Yu, M. Kolbæk, Z.-H. Tan, and J. Jensen, "Permutation invariant training of deep models for speaker-independent multi-talker speech separation," in IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2017, pp. 241–245.

[7] D. Park, W. Chan, Y. Zhang, C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "Specaug- ment: A simple data augmentation method for automatic speech recognition," in IN- TERSPEECH, 2019.

[8] Y. Fujita, S. Watanabe, S. Horiguchi, Y. Xue, and K. Nagamatsu, "End-to-end neu- ral diarization: Reformulating speaker diarization as simple multi-label classification," ArXiv, vol. abs/2003.02966, 2020.

[9] Y. Fujita, S. Watanabe, S. Horiguchi, Y. Xue, and K. Nagamatsu, "End-to-end neu- ral diarization: Reformulating speaker diarization as simple multi-label classification," ArXiv, vol. abs/2003.02966, 2020.

[10] 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," ArXiv, 2020.