Spontaneous Speech Summarization: Transformers All The Way Through

Tomoki Hayashi*, Takenori Yoshimura*, Masaya Inuzuka*, Ibuki Kuroyanagi*, Osamu Segawa†
*Human Dataware Lab. Co., Ltd., Nagoya, Japan
{hayashi,yoshimura.takenori,inuzuka.masaya,kuroyanagi.ibuki}@hdwlab.co.jp
†Chubu Electric Power Co., Inc., Nagoya, Japan
segawa.osamu@chuden.co.jp

Abstract—**This paper proposes a speech summarization system for spontaneous speech. The proposed system consists of speech segmentation, speech recognition, and extractive text summarization modules. We utilize the Transformer architecture for all modules, enabling us to achieve outstanding performance by capturing global and local context information from the sequence thanks to the self-attention mechanism. Furthermore, we introduce a novel data augmentation method for speech summarization using the results of speech segmentation and recognition modules. The proposed data augmentation addresses each sentence boundary's ambiguity in spontaneous speech, making it possible to improve the robustness for speech segmentation and recognition errors. We conduct an experimental evaluation using the Corpus of Spontaneous Japanese, which consists of Japanese speech such as lecture and conference talks. Through the experimental evaluation, we investigate individual performance and each module's relationship in terms of text summarization performance and demonstrate the effectiveness of the proposed data augmentation method.**

Index Terms—**Speech summarization, Transformer, data augmentation, extractive summarization**

I. Introduction

Speech summarization is a technique to summarize long speech as text with a limited number of words. The technique plays an important role in improving document review, video content retrieval, automatic meeting content summarization, etc. With the increase in the demand for online lectures and telecommunication, speech summarization techniques have attracted more attention.

Many researchers have studied speech summarization for various types of speech. The target speech includes telephone dialogue, meeting conversation [1], broadcast news [2] and conference talk [3], [4]. However, since these studies have utilized the manually transcribed text or directly used speech recognition system outputs as the input for the text summarization system, the evaluation of the relationship between speech recognition and text summarization performance has not yet been fully investigated. To reveal the relationship, we build a speech summarization system by cascading speech segmentation, speech recognition, and text summarization modules. These modules are all based on Transformer [5], whose effectiveness has been shown in various domains [6]–[8].

One of the critical problems of using deep learning techniques is the lack of training data. With the improvement of deep learning techniques, various kinds of text summarization methods have been proposed. Sequence-to-sequence (S2S) encoder-decoder-based methods have demonstrated the remarkable performance [9], [10]; however, it generally requires a large amount of training data to achieve a reasonable performance, which is typically difficult to obtain in the speech summarization scenario. To address this issue, the use of pretrained hidden representations based on Transformer [5], [11] has been attracting attentions [12]–[14], which enables us to build the text summarization model by utilizing the pretrained self-supervised Transformer model. Another approach is data augmentation. Y. Liu et al. [15] have introduced the use of multiple noise signals such as Gaussian noise, word drop, etc. A. Magooda et al. [16] have proposed the domain transfer and data synthesis, generating the training data artificially. The back-summarization technique has been proposed in [17], which reverses the process of the summarization. In this paper, we propose a novel data augmentation method for extractive text summarization by utilizing the cascade speech summarization system. The proposed method increases the training data with the results of speech segmentation and speech recognition results, enabling us to address the ambiguity of sentence boundary in spontaneous speech.

The contributions of the paper are summarized as follows:

- We propose a speech summarization system for spontaneous speech, which consists of three modules: 1) speech segmentation module, 2) speech recognition module, and 3) extractive text summarization module. All of the modules are based on Transformer, which enables us to achieve remarkable performance by capturing global and local context information from the sequence thanks to the self-attention mechanism.
- We propose a novel data augmentation method for speech summarization to address the issue of the border ambiguity between utterances in spontaneous speech. The proposed data augmentation utilizes speech segmentation and recognition results to generate additional training data for extractive summarization.
- We conduct an experimental evaluation with the corpus of spontaneous Japanese (CSJ), consisting of the long lecture speech. We investigate each component's performance and reveal the relationship between each compo-

ISBN: 978-9-0827-9706-0

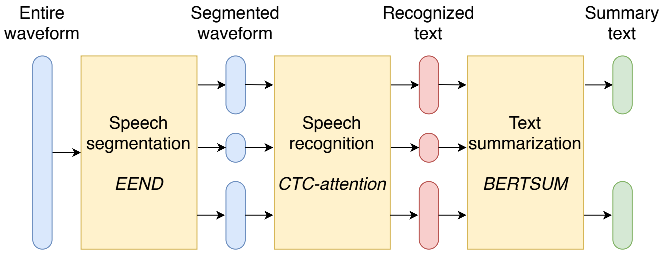

Fig. 1. *An overview of the proposed method.*

nent in terms of text summarization performance. The experimental results also demonstrate the effectiveness of the proposed data augmentation, making it possible to improve the robustness for speech segmentation and speech recognition errors.

II. Method

A. Overview

Fig. 1 shows an overview of the proposed method. The proposed method consists of three Transformer-based modules: 1) speech segmentation module, 2) speech recognition module, and 3) extractive text summarization module.

The speech segmentation module receives unsegmented long speech and split it into each utterance. We use an end-to-end (E2E) neural diarization model called EEND [18] as the speech segmentation module. If the target speech consists of several speakers like an interview, this module works as a speaker diarization. If the target speech includes only a single speaker, such as a lecture, this module works as a simple voice activity detection (VAD).

The speech recognition module receives the segmented utterances and converts each utterance into a sentence. We use an E2E speech recognition model based on the hybrid model of the connectionist temporal classification (CTC) [19] and the attention-based encoder-decoder [20], called joint CTC-attention [21], as a speech recognition module.

The extractive summarization module chooses the important sentences from the text consisting of the recognized sentences. As the extractive summarization module, we use BERTSUM [12], which utilizes the self-supervised learning model BERT [11] for the text summarization. The module performs binary classification of each sentence and constructs the summary to be the desired summary rate. In the following sections, we explain each module in detail.

B. Speech segmentation

The speech segmentation module consists of an input linear layer, a few Transformer blocks, and an output linear layer with a Sigmoid activation function [18]. We extract log Mel-spectrogram from the unsegmented speech. The Mel-spectrogram is spliced with the previous frames and subsequent frames and then subsampled to reduce the length, allowing the long speech to be used as the inputs. The speech segmentation network receives the inputs and performs a binary classification of the target speaker's presence. The number of outputs depends on the type of target speech. For example, if the target speech is like an interview session, the target speakers are two (interviewer and interviewee). If the target speech is the lecture speech, the target speaker is one. The model is trained to minimize the binary cross-entropy with a permutation free loss [22].

C. Speech recognition

The speech recognition module consists of a 2D-convolutional layer, a Transformer encoder, a Transformer decoder, and a CTC branch [6]. The 2D-convolutional layer receives a log Mel-spectrogram extracted from the segmented speech and performs subsampling by stridden convolutions. The subsampled hidden sequence is inputted into the Transformer encoder to get the hidden representation sequence, which considers the input sequence's local and global context. Finally, the decoder and the CTC branch perform decoding with the encoder hidden sequence by a beam-search using a joint CTC-attention decoding [21]. The entire model is optimized with the multi-task learning objective function, which is the sum of the CTC loss function and the attention loss function.

D. Text summarization

The text summarization module consists of the BERT and additional Transformer blocks and a linear layer with a Sigmoid activation function to perform the binary classification [12]. The tokenizer converts the text consisting of the recognized sentences into the token sequence. The special symbols are added to the front and the end of each sentence. We extract three embeddings from the token sequence: 1) token embedding, 2) positional embedding, and 3) interval segment embedding. BERT receives the sum of these embeddings as the inputs and outputs the hidden representation sequence. Then, we extract the hidden representation corresponding to [CLS] and then construct the hidden representation subsequence of [CLS] symbols. The subsequence is inputted into the additional Transformer blocks, and finally, the linear layer with Sigmoid performs a binary classification to determine the important sentence in the text. The model is optimized to minimize the binary cross-entropy.

III. Data Augmentation

The extractive text summarization gives us a concrete summary even with a limited amount of training data; however, it assumes that each sentence in the text has a clear boundary. In general, the target of text summarization is well-formatted text, such as news articles and scientific papers, and therefore, each sentence in the text is separated clearly. On the other hand, our target is spontaneous speech, and therefore the utterance boundary will be ambiguous, affecting the extractive summarization's performance. Furthermore, each sentence might include recognition errors.

To address the above issues, we propose the data augmentation for the extractive text summarization with speech

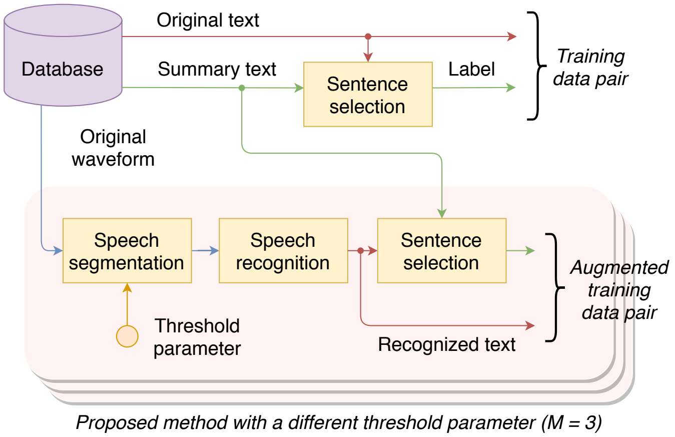

Fig. 2. *The proposed data augmentation procedure.*

recognition results using different speech segmentation results. The proposed data augmentation is illustrated in Fig 2. First, we generate speech segments with various lengths by the speech segmentation module with different M threshold parameters. Second, we recognize these segments by the speech recognition module, obtaining the text consisting of different length sentences. Then, we decide the important sentences by the greedy selection, which chooses the sentences to maximize the ROUGE score [23] using the ground-truth summary text. Finally, we use both the original text and recognized text with different segmentation results as the training data for the extractive summarization, increasing $M+1$ times larger than the original training data. It is expected that this data augmentation enables us to improve the robustness to recognition errors and various speech segmentation results.

IV. Experimental Evaluation

A. Experimental condition

We conducted an experimental evaluation using the corpus of spontaneous Japanese (CSJ) [24]. The dataset consists of the training set, including 3,212 spontaneous speech of lecture and conference talks, and three evaluation sets (eval1, eval2, and evel3), each of which includes ten spontaneous speech. All speech was recorded with 16 kHz and 16 bit. Each spontaneous speech includes the ground-truth of segmentation information and text, but only 168 speech of the training set and the evaluation sets include the ground-truth summary. We used 3,212 speech to train speech segmentation and the speech recognition modules and 168 speech to train extractive text summarization module. The ground-truth summary includes two kinds of summaries with different summary rates: 10% and 50%. Each summary was created by several annotators, including 3-4 kinds of summarized text. We used the summary of 10% summary rate as the ground-truth summary. Since the evaluation sets consist of a single speaker's spontaneous speech, the number of target speakers in the speech segmentation was set to one. The training condition of each module is as follows:

Speech segmentation: To build the speech segmentation module, we used an open-source repository EEND[1]. We extracted 23-dimensional log Mel-spectrogram spliced with the previous and subsequent seven frames and then subsampled by a factor of ten, resulting in 345-dimensional inputs with the length of $T/10$, where T represents the number of the original frames. From the preliminary experiments, we decided to use the same network parameters as those of the original paper [18]: the number of the Transformer blocks and attention heads were set to two and four, respectively. The network parameters were optimized by the Adam algorithm with the warm-up scheduler [5]. The warm-up steps were set to 25,000. After thresholding the network output representing posterior probabilities of speech/non-speech, an 11-th order median filter was applied to the obtained binary sequence to prevent generating very short segments.

Speech recognition: To construct the speech recognition module, we used an open-source E2E speech processing toolkit ESPnet [25]. We followed the CSJ recipe of ESPnet2 (egs2/csj/asr1). The input was an 80-dimensional Mel-spectrogram. The number of encoder blocks, decoder blocks, attention heads were set to 18, six, and eight, respectively. The optimizer setting was the same as the speech segmentation module. To boost up the recognition performance, we used the speed perturbation with the factor of 0.9 and 1.1 and SpecAug [26] during the training. We used joint CTC-attention training and decoding, and the alpha of multi-task learning and CTC weight in the decoding were set to 0.3.

Text summarization: We used the pretrained BERT-base model trained with Japanese wikipedia[2]. The tokenizer was MeCab with the IPA dictionary, followed by SentencePiece [27]. For the additional Transformer encoder, the number of the blocks and their attention heads were set to two and eight, respectively. The network was optimized with the same optimizer and scheduler as the other modules with a smaller learning rate of 0.05 and shorter warm-up steps of 10,000. Since the length of positional embedding in the pretrained BERT was limited to 512, we split the long text into a set of chunked text. In the inference, we sort scores of all sentences in the text and then select the important sentences from the top not to exceed 10% of the number of sentences in the text.

B. Experimental results

First, we focus on the results of speech segmentation and speech recognition modules. Table I summarizes the speech segmentation and speech recognition results, where "Threshold" represents the threshold value of the speech segmentation module, which is in the range of $[0, 1]$, "DER" represents the diarization error rate, "CER" represents the character error rate, and "Length" represents the average length of segmented

[1] https://github.com/hitachi-speech/EEND
[2] https://github.com/cl-tohoku/bert-japanese

TABLE I
Experimental results of speech segmentation and speech recognition. The successive values in "DER" and "CER" represent the score for eval1, eval2, and eval3 sets, respectively. The values in "Length" represent the mean and standard deviation of the speech segments.

Threshold	DER [%]	CER [%]	Length [sec]
GT	N/A	4.9 / 3.7 / 3.9	N/A
0.1	1.9 / 1.3 / 1.7	**6.3 / 4.3 / 4.7**	8.1 ± 7.0
0.2	1.6 / 1.1 / 1.3	6.4 / 4.4 / 5.0	7.4 ± 6.6
0.3	1.5 / 1.0 / 1.1	6.5 / 4.4 / 5.1	7.0 ± 6.4
0.4	1.5 / 0.9 / 0.8	6.5 / 4.5 / 5.3	6.7 ± 6.2
0.5	1.5 / 0.8 / 0.8	6.6 / 4.6 / 5.5	6.4 ± 6.1
0.6	**1.5 / 0.7 / 0.8**	6.7 / 4.7 / 5.7	6.2 ± 5.9
0.7	1.5 / 0.8 / 0.8	6.8 / 4.9 / 5.9	6.0 ± 5.7
0.8	1.6 / 0.8 / 1.1	7.0 / 5.0 / 6.3	5.7 ± 5.4
0.9	3.1 / 9.5 / 12.8	8.3 / 12.4 / 19.5	4.7 ± 4.4

TABLE II
Experimental results of the text summarization module.

Method	R-1	R-2	R-3	R-L
Oracle	62.5	39.0	29.5	45.6
BERTSUM	45.0	18.2	11.2	**21.9**
+ GT seg. + SR	43.6	16.2	9.4	**21.0**
+ SS seg. (0.1) + SR	42.9	16.2	9.1	**21.8**
+ SS seg. (0.3) + SR	44.0	16.3	9.3	21.6
+ SS seg. (0.5) + SR	45.1	17.1	10.3	20.6
+ SS seg. (0.7) + SR	42.6	14.8	8.2	20.3
+ SS seg. (0.9) + SR	42.7	16.0	9.3	20.1
ALL TR. (Ours)	44.0	16.5	9.4	20.4
+ GT seg. + SR	42.3	14.9	8.1	19.9
+ SS seg. (0.1) + SR	42.1	14.6	8.2	20.1
+ SS seg. (0.3) + SR	42.9	15.6	9.0	20.5
+ SS seg. (0.5) + SR	43.5	15.4	8.4	20.6
+ SS seg. (0.7) + SR	**43.8**	16.2	9.2	**20.7**
+ SS seg. (0.9) + SR	**43.1**	15.6	8.7	**20.5**
ALL TR. + AUG (Ours)	**45.5**	**19.7**	**12.6**	21.8
+ GT seg. + SR	**45.1**	**18.5**	**11.1**	20.7
+ SS seg. (0.1) + SR	**43.9**	**17.3**	**10.5**	21.2
+ SS seg. (0.3) + SR	**45.0**	**18.4**	**11.6**	**21.9**
+ SS seg. (0.5) + SR	**45.2**	**18.5**	**11.4**	**20.8**
+ SS seg. (0.7) + SR	**43.8**	**17.8**	**10.9**	20.5
+ SS seg. (0.9) + SR	42.2	**16.4**	**9.9**	19.9

utterances with the standard deviation. The results showed that using a higher threshold in the speech segmentation module made the shorter segments, resulting in a larger number of sentences from the input speech. On the other hand, using a lower threshold in the speech segmentation module made the longer segments, resulting in a smaller number of sentences. In terms of DER, around 0.5 is the best for the threshold, but a lower threshold (i.e., longer segments) brought better CER than a higher threshold. One of the reasons for this behavior is that the speech recognition module can use more context information in the sentence if the longer segments are provided. However, since the self-attention module requires the computational const $\mathcal{O}(N^2)$, where N is the sequence length, the use of longer segments made the decoding slower.

Next, we focus on the results of text summarization. To check the effectiveness of the proposed data augmentation method, we compared the following models:

Oracle: The oracle score calculated by the greedy selection with ground-truth summaries.

BERTSUM: The BERTSUM [12] model trained with only the original training data (the ground-truth text).

All TR.: The proposed all Transformer model trained with the recognized text produced by the speech segmentation $(\mathrm{threshold} = 0.5)$ and the speech recognition modules instead of the ground-truth text.

All TR. + AUG: The proposed model trained with the proposed data augmentation method. We used $M = 9$ for the data augmentation with threshold values from 0.1 to 0.9, resulting in ten times larger than the original training data.

The other training conditions were the same among the above models. Table II shows the text summarization results, where "R-1", "R-2", "R-3", and "R-L" represent ROUGE-1, ROUGE-2, ROUGE-3, and ROUGE-L F-measure score, respectively. ROUGE-N ($N = \{1, 2, 3\}$) F-measure represents an F-measure of N-grams between the output and reference summaries and ROUGE-L represents that of the longest matching sequence of words in the summary [23]. The first row in each segment represents the score using ground-truth text. "+ GT seg. + SR" represents the score using the recognized text for the inference instead of the ground-truth text. The recognized text was created by using the ground-truth speech segments and the speech recognition module. "+ SS seg. + SR" also represents the score using the recognized text of those segments predicted by the speech segmentation module. The value in parentheses represents the threshold value of the speech segmentation module.

From the results in Table II, ALL TR. (Ours) gave a better performance than BERTSUM in the higher threshold conditions but worse in the cases of lower threshold or using ground-truth. Since Table I shows the higher threshold gave worse CER, ALL TR. assumed the existence of recognition errors, and therefore, it degraded the performance for the ground-truth text, which does not include recognition errors. On the other hand, ALL TR. + AUG (Ours) brought consistent improvement for most of the input conditions, improving the basic performance and the robustness to the recognition errors.

However, there was a big gap between the oracle score. The main reason for this big gap was the limited amount of training data. Even if we used the proposed data augmentation method, the number of the training text is less than 2,000, much smaller than the dataset for text summarization such as CNN daily mail [28]. To address this issue, we will consider unsupervised data augmentation based on information criterion or graph analysis to utilize a large amount of text without a summary for extractive summarization in future work.

Focusing on the difference of threshold values in ALL TR. + AUG (Ours), the lower threshold value (i.e., longer sentences) did not always improve the ROUGE score, even if the CER was better. This implied that there is a suitable sentence length for extractive summarization, and we have room for improvement by feedback from the text summarization output to the speech segmentation module. We will consider the feedback method in future work.

V. Conclusion

This paper proposed the Transformer-based speech summarization method for spontaneous speech. To address the boundary ambiguity issue, we proposed a novel data augmentation method using the results of speech segmentation and speech recognition modules. From the experimental results with CSJ, we investigated the relationship of each component's performance and revealed the effectiveness of the proposed data augmentation method in various input conditions. In future work, we will consider each module's joint training, the data augmentation method using the training data without the ground-truth summary, and investigate the performance on the other types of the dataset.

References

[1] Chenguang Zhu, Ruochen Xu, Michael Zeng, and Xuedong Huang, "A hierarchical network for abstractive meeting summarization with cross-domain pretraining," *arXiv preprint arXiv:2004.02016*, 2020.

[2] Kuan-Yu Chen, Shih-Hung Liu, Berlin Chen, Hsin-Min Wang, Ea-Ee Jan, Wen-Lian Hsu, and Hsin-Hsi Chen, "Extractive broadcast news summarization leveraging recurrent neural network language modeling techniques," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 8, pp. 1322–1334, 2015.

[3] Sadaoki Furui, Tomonori Kikuchi, Yosuke Shinnaka, and Chiori Hori, "Speech-to-text and speech-to-speech summarization of spontaneous speech," *IEEE Transactions on Speech and Audio Processing*, vol. 12, no. 4, pp. 401–408, 2004.

[4] Derek Miller, "Leveraging BERT for extractive text summarization on lectures," *arXiv preprint arXiv:1906.04165*, 2019.

[5] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin, "Attention is all you need," in *Proc. Advances in Neural Information Processing Systems*, 2017, pp. 5998–6008.

[6] Shigeki Karita, Nanxin Chen, Tomoki Hayashi, Takaaki Hori, Hirofumi Inaguma, Ziyan Jiang, Masao Someki, Nelson Enrique Yalta Soplin, Ryuichi Yamamoto, Xiaofei Wang, et al., "A comparative study on Transformer vs RNN in speech applications," in *Proc. Automatic Speech Recognition and Understanding Workshop*. IEEE, 2019, pp. 449–456.

[7] Wen-Chin Huang, Tomoki Hayashi, Yi-Chiao Wu, Hirokazu Kameoka, and Tomoki Toda, "Voice transformer network: Sequence-to-sequence voice conversion using transformer with text-to-speech pretraining," *arXiv preprint arXiv:1912.06813*, 2019.

[8] Koichi Miyazaki, Tatsuya Komatsu, Tomoki Hayashi, Shinji Watanabe, Tomoki Toda, and Kazuya Takeda, "Weakly-supervised sound event detection with self-attention," in *Proc. International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2020, pp. 66–70.

[9] Zhengyuan Liu, Angela Ng, Sheldon Lee, Ai Ti Aw, and Nancy F Chen, "Topic-aware pointer-generator networks for summarizing spoken conversations," in *Proc. Automatic Speech Recognition and Understanding Workshop*. IEEE, 2019, pp. 814–821.

[10] Xinyuan Zhang, Ruiyi Zhang, Manzil Zaheer, and Amr Ahmed, "Unsupervised abstractive dialogue summarization for tete-a-tetes," *arXiv preprint arXiv:2009.06851*, 2020.

[11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, "BERT: Pre-training of deep bidirectional Transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[12] Yang Liu, "Fine-tune BERT for extractive summarization," *arXiv preprint arXiv:1903.10318*, 2019.

[13] Xingxing Zhang, Furu Wei, and Ming Zhou, "HIBERT: Document level pre-training of hierarchical bidirectional transformers for document summarization," *arXiv preprint arXiv:1905.06566*, 2019.

[14] Shashi Narayan, Joshua Maynez, Jakub Adamek, Daniele Pighin, Blaž Bratanič, and Ryan McDonald, "Stepwise extractive summarization and planning with structured transformers," *arXiv preprint arXiv:2010.02744*, 2020.

[15] Yang Liu, Sheng Shen, and Mirella Lapata, "Noisy self-knowledge distillation for text summarization," *arXiv preprint arXiv:2009.07032*, 2020.

[16] Ahmed Magooda and Diane Litman, "Abstractive summarization for low resource data using domain transfer and data synthesis," *arXiv preprint arXiv:2002.03407*, 2020.

[17] Paul Tardy, Louis de Seynes, François Hernandez, Vincent Nguyen, David Janiszek, and Yannick Estève, "Leverage unlabeled data for abstractive speech summarization with self-supervised learning and back-summarization," in *Proc. International Conference on Speech and Computer*. Springer, 2020, pp. 572–580.

[18] Yusuke Fujita, Naoyuki Kanda, Shota Horiguchi, Yawen Xue, Kenji Nagamatsu, and Shinji Watanabe, "End-to-end neural speaker diarization with self-attention," in *Proc. Automatic Speech Recognition and Understanding Workshop*. IEEE, 2019, pp. 296–303.

[19] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proc. International Conference on Machine Learning*, 2006, pp. 369–376.

[20] Jan K Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio, "Attention-based models for speech recognition," in *Proc. Advances in Neural Information Processing Systems*, 2015, pp. 577–585.

[21] Shinji Watanabe, Takaaki Hori, Suyoun Kim, John R Hershey, and Tomoki Hayashi, "Hybrid CTC/attention architecture for end-to-end speech recognition," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 8, pp. 1240–1253, 2017.

[22] John R Hershey, Zhuo Chen, Jonathan Le Roux, and Shinji Watanabe, "Deep clustering: Discriminative embeddings for segmentation and separation," in *Proc. International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2016, pp. 31–35.

[23] Chin-Yew Lin, "ROUGE: A package for automatic evaluation of summaries," in *Text Summarization Branches Out*, Barcelona, Spain, July 2004, pp. 74–81, Association for Computational Linguistics.

[24] Kikuo Maekawa, "Corpus of Spontaneous Japanese: Its design and evaluation," in *Proc. Workshop on Spontaneous Speech Processing and Recognition*. ISCA & IEEE, 2003.

[25] Shinji Watanabe, Takaaki Hori, Shigeki Karita, Tomoki Hayashi, Jiro Nishitoba, Yuya Unno, Nelson Enrique Yalta Soplin, Jahn Heymann, Matthew Wiesner, Nanxin Chen, et al., "ESPnet: End-to-end speech processing toolkit," *arXiv preprint arXiv:1804.00015*, 2018.

[26] Daniel S Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D Cubuk, and Quoc V Le, "SpecAugment: A simple data augmentation method for automatic speech recognition," *arXiv preprint arXiv:1904.08779*, 2019.

[27] Taku Kudo and John Richardson, "SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing," *arXiv preprint arXiv:1808.06226*, 2018.

[28] Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom, "Teaching machines to read and comprehend," in *Proc. Advances in Neural Information Processing Systems*, 2015, pp. 1693–1701.