Real-time VAD-less speech recognition by fine-tuning SSL model with data containing tagged non-speech segments

Jotaro Emoto*, Ryota Nishimura†, Kengo Ohta‡ and Norihide Kitaoka†
* Tokushima University, Japan
E-mail: c612435044@tokushima-u.ac.jp
† Toyohashi University of Technology, Japan
E-mail: nishimura.ryota.tz@tut.jp, kitaoka@tut.jp
‡ National Institute of Technology Anan College, Japan
E-mail: kengo@anan-nct.ac.jp

***Abstract*—Real-time speech recognition is crucial in systems requiring quick responses, such as spoken dialogue systems and other interactive applications. However, background noise significantly reduces recognition accuracy in real-world environments. Voice activity detection (VAD) is a common countermeasure, but it has significant drawbacks such as false detections and a fundamental lack of real-time performance due to its reliance on future audio segments. In this study, we develop a real-time speech recognition model capable of simultaneously processing both speech and non-speech segments. This is achieved by CTC fine-tuning a self-supervised learning (SSL) model using speech data where non-speech segments are explicitly tagged in the transcription. Our experimental results demonstrate that for speech rich in non-speech intervals that would typically require VAD, our proposed model achieves VAD-like behavior in handling these non-speech segments. This approach led to a recognition performance increase of up to 9% compared to conventional VAD methods in noisy, real-world scenarios.**

I. Introduction

Recent developments in deep learning technology and the release of large-scale datasets have improved the performance of speech recognition systems. In particular, Self-Supervised Learning (SSL) models such as wav2vec 2.0 [1] and HuBERT [2] enable the creation of high-accuracy speech recognition models. This is achieved by pre-training them with extremely large amounts of speech data, followed by fine-tuning with smaller amounts of data from the target domain. This method is very effective when implementing automatic speech recognition (ASR) for low-resource languages [3]–[6]. It has also been demonstrated that increasing the amount of speech data used during pre-training and fine-tuning improves the performance of the final model [7]. Thus, it is presumed that utilizing more target-domain data during fine-tuning will further enhance recognition accuracy and improve domain-specificity.

However, the issue of accuracy degradation due to background noise remains. One measure used to reduce noise-related degradation of speech recognition performance is Voice Activity Detection (VAD). VAD functions by removing the non-speech segments from the input speech signal and then feeding only speech into the speech recognition model, thereby improving recognition accuracy. Furthermore, it has been reported that incorporating background noise into the speech used for pre-training and fine-tuning of the SSL model also enhances speech recognition accuracy [8].

However, if VAD incorrectly removes a speech segment, a non-speech segment will be input into the subsequent speech recognition module, which can have an irreversible, negative effect on the speech recognition results. Moreover, since the VAD requires access to the audio signal after the targeted speech segment, it cannot be applied in real-time speech recognition systems, which, by definition, can only process past information. Therefore, attempts have been made to integrate VAD into ASR. For instance, Yoshimura et al. [9] utilized the length of consecutive blank tokens (i.e., non-speech segments) within Connectionist Temporal Classification (CTC) [10], [11]. Sashi et al. [12] combined the output probability of VAD with the input features of the ASR model, and distilled the knowledge of VAD into the ASR model through multi-task learning. Improvements in recognition accuracy were reported in these studies, but details regarding their real-time processing capabilities or the specific handling of speech segments were not provided.

In this study, we propose a real-time speech recognition model that achieves VAD-like behavior by training the speech recognition model with audio data into which non-speech segments with noise have been inserted, and by adding non-speech tags to the corresponding segments in the transcription. The results of our speech recognition experiment showed a slight decrease in accuracy for speech that did not require VAD (i.e., speech without many non-speech intervals); however, a significant increase in accuracy was observed for data rich in non-speech intervals, which would have significantly benefited from VAD. In addition, it was observed that an online speech recognition system which did not use a GPU could process two seconds of audio in approximately 120 ms when using our proposed method.

The contributions of our research are as follows:

979-8-3315-7206-8/25/$31.00

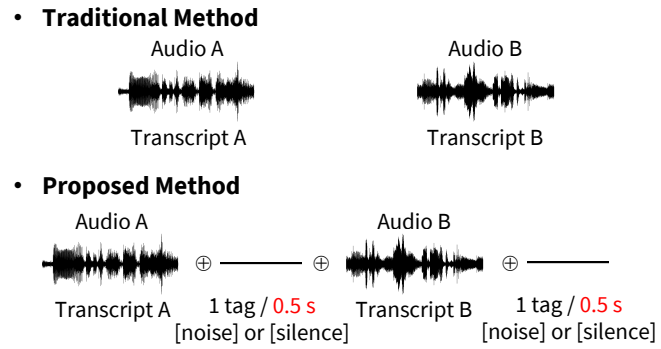

Fig. 1. Schematic of proposed method for preparing additional training data for the wav2vec 2.0 model. Two audio segments are combined and two non-speech segments are inserted.

1) Development of a speech recognition model that inherently achieves VAD-like behavior, enabling real-time processing with only 120 ms latency on a CPU under various noise conditions.
2) A simple method for fine-tuning the CTC of a SSL model, using speech data into which non-speech segments have been inserted, which are then explicitly tagged in the transcription data.
3) Extensive experimental evaluation demonstrating a relative improvement in ASR performance of up to 9% when processing noisy data compared to conventional VAD-based approaches.

II. Proposed Method

A. Connectionist Temporal Classification (CTC)

Connectionist Temporal Classification (CTC) [10], [11] is a method employed in sequence-to-sequence tasks, such as time series prediction. In speech recognition, it is commonly used with encoder-only transformer models. A fully connected layer, producing an output of an arbitrary vocabulary size, is added to the final layer of the model, outputting the probability of each token per frame.

One of the features of CTC is that it outputs blank tokens to represent frames that do not match any vocabulary. Blank tokens output are not only for the boundaries of unknown words, but also for the boundaries of non-speech segments. In this study, we focused on the blank tokens output for these non-speech segments. By utilizing blank tokens to represent non-speech segments, we can construct a speech recognition model capable of indirectly detecting potentially noisy, non-speech intervals concurrently with ASR processing.

B. wav2vec 2.0

When constructing our ASR model, we adopted wav2vec 2.0 [1] as the base SSL model. It extracts audio features 25 ms in length every 20 ms. Consequently, by training it using CTC, recognition results can be obtained every 20 ms. Therefore, if a segment contains speech, the recognition result is obtained every 20 ms, but if it is a non-speech segment, a blank token is obtained.

In terms of architecture, we adopted a wav2vec 2.0 base model with 90M parameters. This choice was made to minimize the computational cost of the ASR model, thereby enabling real-time operation. The pre-trained wav2vec 2.0 model used in this study is described in more detail in the Experiments section.

C. Data Preparation

To train the proposed model, it is essential to include non-speech segments in the audio data. However, as existing corpora often lack a sufficient number of non-speech segments, the training data was created according to the following procedure. An outline of the data preparation is shown in Figure 1.

1) Insertion of a non-speech segments: Since it is not always possible to insert non-speech segments into all speech, and because forcibly dividing utterances in order to insert non-speech segments may damage natural pronunciation and negatively affect the performance of the final speech recognition model, we used the two audio segments as a single data. When creating the proposed model, we used data with a non-speech segment of 3 to 5 seconds between the two speech segments, and a non-speech segment of 1 or 2 seconds at the end.

2) Tagging of non-speech segments: To enable the speech recognition model to recognize the inserted non-speech segments, we tagged their corresponding transcription intervals. As shown in Figure 1, we inserted tags in the middle of the transcription, corresponding to the period from the end of the first speech segment until the beginning of the second speech segment, and then from the end of the second speech segment until the end of the transcription.

Additionally, two types of tags were inserted, according to the level of noise added to the audio. If the Signal-to-Noise Ratio (SNR) of the added noise was less than 20 dB, the resulting non-speech section was considered noisy, and thus a "[noise]" tag was inserted. If the SNR was greater than that, a "[silence]" tag was inserted. It should be noted that these inserted tags were added to enable the model to explicitly distinguish and output non-speech segments. Therefore, they do not carry any linguistic meaning, and any character string could be used for this purpose.

III. Experiments

We conducted speech recognition experiments in English and Japanese to confirm the influence of linguistic features. In this section, we will explain the pre-trained models and datasets used in the experiment and the types of models used for evaluation.

A. Pre-trained Models

Two pre-trained wav2vec 2.0 models were used in our experiments. The **facebook/wav2vec2-base**[1] model was pre-trained using 960 hours of English speech from the LibriSpeech [13] dataset, whereas the **rinna/japanese-wav2vec2-**

[1]https://huggingface.co/facebook/wav2vec2-base

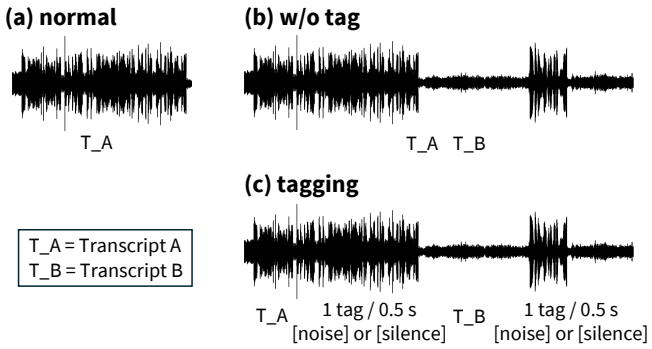

Fig. 2. Schematic of data used for model training. Proposed method is '(c) tagging'. For methods (b) and (c), two audio segments are combined and two non-speech segments are inserted.

base[2] model, highly specialized for Japanese, was pre-trained with 19,000 hours of Japanese speech from the ReazonSpeech v1.0 [14] dataset.

B. Training Datasets

The LibriSpeech [13] dataset used for training the English recognition model consists of audio book recordings, with a total recording time of 960 hours. We used the train-clean-100 subset, which contains 100 hours of data, for training the wav2vec 2.0 model.

ReazonSpeech v2.0 [14] consists of audio data collected from one-segment TV broadcasts, totaling 35,000 hours of recording time. Five subsets (tiny, small, medium, large, and all) have been released, which are named according to the amount of data they contain. In this experiment, we used the 'small' subset to train the Japanese wav2vec 2.0 model, which contains the same 100 hours of audio data as the LibriSpeech dataset used to train the English recognition model. The ReazonSpeech dataset contains a mixture of spontaneous speech and read speech, and contains a relatively high proportion of noisy speech.

C. Noise Addition

The additional noise used in the experiments was babble noise and pink noise from NOISEX-92 [15]. The babble noise in this dataset was recorded in a dining hall with a radius of 2 meters or more, featuring 100 people talking. The pink noise (which sounds like a waterfall) was applied at SNR = 50 dB to serve as a noise floor in the input audio, while the babble noise was added in three stages of SNR = 0 dB, 5 dB, and 10 dB, to mimic real-world environments. The noise in the dataset is publicly available, sampled at 19,980 Hz. Therefore, each noise sample was resampled to 16,000 Hz for input into the wav2vec 2.0 models. Note that the RMS used for SNR calculation excludes non-speech segments, and noise was added to the entire audio, including the non-speech segments. While this study uses two specific noise types to examine VAD-like behavior in clean and noisy environments, we acknowledge that evaluating with a broader range of noise types would provide more comprehensive insights into our approach's robustness.

[2]https://huggingface.co/rinna/japanese-wav2vec2-base

D. Training of Models for Experiment

To observe the effects of inserting non-speech segments into the training data and adding explicit non-speech tags (e.g., [noise], [silence]) to the vocabulary, we prepared three types of training data and trained speech recognition models for each experiment. An overview of the data used to train each model is shown in Figure 2.

- **normal**
  Model trained with all audio data, including added noise.
- **w/o tag**
  Model trained with audio data with non-speech segments inserted, but no additional tags are used to identify locations of non-speech segments, in the transcription or vocabulary.
- **tagging**
  Model trained with audio data into which non-speech segments were inserted, with their locations tagged in the transcription data.

E. Training Setup

In addition to 'unknown vocabulary' tags ("<unk>"), three other special tokens are included in the model's vocabulary during training; "[PAD]", "<s>", and "</s>". When training the English ASR model, the transcription is encoded using characters, so the vocabulary size is 34 after adding the proposed tags. When training the Japanese ASR model, the base vocabulary (size 3,200, including the "<unk>" tag) was derived from a unigram [16] language model trained with SentencePiece [17]. After incorporating the proposed non-speech tags and other special tokens, the final vocabulary size was 3,205.

During fine-tuning, AdamW with $\beta = (0.9, 0.999)$ was used as the optimization function. Training was conducted for 20 epochs, with the learning rate increasing to a maximum of $5e-5$ for the first 10% of training (warm-up), after which it was gradually reduced using a cosine annealing schedule. The batch size was 4, the gradient accumulation was 2, and the execution batch size was 8. In addition, as indicated in the paper [1], the feature extraction layer's CNN was frozen. The time required for fine-tuning was about 3 hours using one Nvidia RTX 6000 Ada graphics card.

F. Model Evaluation

The models were evaluated using two types of data: the original corpus speech (which does not include many non-speech segments), and speech data augmented using the proposed method. For the evaluation of the English ASR model, we used the test-clean and test-other subsets from the LibriSpeech corpus, and for the evaluation of the Japanese model, we used the JNAS [18], Common Voice 8.0 [19], and ReazonSpeech [14] corpora. We also calculated recognition results for speech that included non-speech segments, where VAD was applied as

TABLE I
COMPARISON OF ENGLISH ASR MODEL PERFORMANCE USING AUDIO DATA THAT DOES NOT CONTAIN NON-SPEECH SEGMENTS. EVALUATION WAS CONDUCTED USING LIBRISPEECH'S TEST-CLEAN AND TEST-OTHER.

Model	test-clean						test-other					
	clean	20	15	10	5	0	clean	20	15	10	5	0
base.en	2.22	2.39	2.54	3.42	6.87	23.45	5.73	6.35	8.49	10.03	18.19	44.34
v3-turbo	**1.61**	**1.46**	**1.51**	**1.64**	**2.47**	**7.15**	**2.54**	**2.64**	**2.95**	**3.77**	**6.70**	**18.36**
normal	3.52	3.64	3.78	4.30	6.16	13.39	6.87	7.58	8.31	10.34	15.65	27.94
w/o tag	3.74	3.86	4.07	4.55	6.67	14.55	7.22	7.96	8.91	11.08	16.79	29.71
tagging(proposed)	3.75	3.87	4.11	4.72	6.82	14.98	7.37	8.16	9.11	11.42	17.16	30.19

TABLE II
PERFORMANCE COMPARISON OF ENGLISH ASR MODELS ON LIBRISPEECH TEST-CLEAN SUBSET INCLUDING NON-SPEECH SEGMENTS.

	CER under SNR(dB)					
	clean	20	15	10	5	0
Baselines:						
base.en	13.92	6.82	7.38	7.47	11.75	32.91
v3-turbo	6.91	6.21	6.67	5.68	**7.15**	**15.35**
Our Models (Training Methods):						
normal	4.43	4.65	4.99	5.90	8.84	18.61
w/o tag	4.07	4.21	4.41	5.08	7.54	16.14
tagging(proposed)	**3.75**	**4.01**	**4.19**	**4.86**	7.21	16.04
VAD Preprocessing Evaluation:						
normal + text_c	4.31	5.23	5.84	6.05	8.25	17.92
normal + seg_c	4.84	5.14	5.45	6.18	8.85	18.59

a preprocessing step. To compare with VAD-based approaches, we applied **pyannote/voice-activity-detection**[3] [20], [21], in conjunction with our 'normal' model and evaluated two VAD processing methods:

- **text_c**
  Speech recognition is performed on each VAD-detected segment separately, and the results are then concatenated.
- **seg_c**
  VAD-detected segments are concatenated first, and speech recognition is then performed on the combined audio.

In addition to the models we created, we also tested the following publicly available models.

- **reazon-research/japanese-wav2vec2-base-rs35kh**[4]
  This model is based on the wav2vec 2.0 base architecture, pre-trained on 35,000 hours of Japanese speech data from ReazonSpeech v2.0, and subsequently fine-tuned on the same dataset for CTC.
- **whisper** [22]
  We used the **whisper-base**[5] (74M) model, whose parameter count is similar to that of the wav2vec 2.0 base model, and the **whisper-large-v3-turbo**[6] (809M) model, which was the most recent model at the time of writing. For the English recognition experiment, we used the **whisper-base.en**[7] model, which only recognizes English.

We used the Character Error Rate (CER) to evaluate the models. We removed all tags from the recognition results, employed greedy decoding, and did not use any additional language models. To assess performance speed, we also measured the inference speed of each Japanese ASR model using Real-Time Factor (RTF), with the JNAS dataset. Since the wav2vec 2.0 model supports online inference, we also measured the latency for recognition during online use. The GPU used for the measurements was an RTX3090, and we also calculated the results using a Macbook Air (M1, 8-core CPU).

[3]https://huggingface.co/pyannote/voice-activity-detection
[4]https://huggingface.co/reazon-research/japanese-wav2vec2-base-rs35kh
[5]https://huggingface.co/openai/whisper-base
[6]https://huggingface.co/openai/whisper-large-v3-turbo
[7]https://huggingface.co/openai/whisper-base.en

IV. RESULTS

A. English ASR performance

Results for the English ASR models are shown in Table I (without non-speech segments) and Table II (with non-speech segments). When non-speech segments were not included, the whisper-large-v3-turbo model achieved the best performance in all cases. On the other hand, when non-speech segments are included, the Whisper model experiences significant accuracy degradation, while our proposed 'tagging' model achieved the best performance with clean data and in environments with SNR levels above 10 dB. Comparing the test-clean in Table I and Table II, the whisper model shows significant accuracy degradation when non-speech segments are present, while our proposed model maintains more stable performance. This suggests that the proposed method enables the speech recognition model to properly interpret non-speech segments, thereby obviating the need for VAD preprocessing. At higher environmental noise levels, our model's performance is comparable to that of the best-performing model, and we believe its performance can be further improved by increasing the amount of training data. In addition, when comparing the models we trained for this experiment, the 'tagging' model achieved the best performance, demonstrating a relative accuracy improvement of up to 21%.

B. Japanese ASR performance

The results for the Japanese ASR models are shown in Table III (without non-speech segments) and Table IV (with non-speech segments). The recognition accuracy of whisper-large-v3-turbo was generally good, but when recognizing the ReazonSpeech data, which does not contain non-speech segments, the wav2vec 2.0 'rs35kh' model achieved better accuracy. When comparing models trained with non-speech segments, the 'tagging' model achieved better accuracy than both VAD-based approaches and the 'normal' baseline model. However, the 'w/o tag' model, trained on the same audio data but without explicit transcription tags, unexpectedly outperformed the 'tagging' model in this specific case. However, when the Facebook-pretrained model was used, the 'tagging' model consistently yielded the best accuracy. These results are discussed in detail in Section V.

TABLE III
COMPARISON OF JAPANESE ASR MODELS WITHOUT NON-SPEECH SEGMENTS.

Model	JNAS						Common Voice 8.0						ReazonSpeech					
	clean	20	15	10	5	0	clean	20	15	10	5	0	clean	20	15	10	5	0
base	29.11	26.03	25.15	28.69	34.92	94.85	29.31	31.61	36.27	46.62	82.88	207.89	72.23	75.22	77.79	91.08	120.94	194.22
v3-turbo	**6.98**	**6.92**	**7.07**	**7.24**	**8.43**	**14.88**	**9.48**	**9.89**	**10.01**	**13.06**	**20.25**	48.33	12.72	13.29	14.29	15.99	21.25	44.65
rs35kh	10.03	9.34	9.94	11.18	15.74	32.61	15.64	16.43	18.25	21.88	30.77	50.21	**12.52**	**12.96**	**13.55**	**14.99**	**19.56**	34.35
normal	17.41	17.23	17.52	18.21	19.97	26.68	20.88	21.27	22.03	24.04	29.03	40.99	19.20	19.23	19.39	20.08	22.10	**28.31**
w/o tag	17.61	17.55	17.69	18.13	19.94	26.26	21.26	21.67	22.60	24.46	29.05	40.68	21.30	21.44	21.65	22.37	24.29	30.21
tagging	18.35	18.23	18.53	18.71	20.07	26.67	21.94	22.38	23.20	25.23	29.21	**39.61**	21.52	21.43	21.62	22.29	24.11	29.20

TABLE IV
PERFORMANCE COMPARISON OF JAPANESE ASR MODELS ON JNAS TEST SET INCLUDING NON-SPEECH SEGMENTS.

	CER under SNR(dB)					
	clean	20	15	10	5	0
Baselines:						
base	31.58	28.43	31.04	32.40	46.29	93.71
v3-turbo	**8.07**	**7.54**	**8.14**	**9.53**	**13.59**	**27.19**
Our Models (Rinna Pretrained Model):						
normal	21.88	22.08	22.87	24.64	27.79	37.64
w/o tag	17.81	18.02	18.53	20.10	23.54	33.86
tagging(proposed)	18.36	20.16	19.62	20.14	23.40	32.00
VAD Preprocessing Evaluation:						
normal + text_c	28.16	27.53	27.46	28.82	30.96	38.10
normal + seg_c	33.40	32.36	33.15	33.34	35.27	41.29
Our Models (Facebook Pretrained Model):						
normal	31.68	33.59	33.57	35.89	41.13	53.85
w/o tag	25.73	26.59	27.47	30.21	36.52	51.06
tagging(proposed)	24.19	24.87	26.15	28.51	35.26	49.52
VAD Preprocessing Evaluation:						
normal + text_c	35.96	35.96	36.89	38.89	42.63	53.46
normal + seg_c	36.62	36.11	37.38	40.02	45.39	56.01

C. Inference Speed

As shown in Table V, when using a GPU for inference with the Japanese ASR models, the wav2vec 2.0 model demonstrated the fastest performance in standalone operation. Although their parameters were similar, it was approximately 10 times faster than the whisper-base model and about twice as fast when used with VAD. We also measured the latency when using the wav2vec 2.0 model online. We performed microphone input and model inference simultaneously, running the system as fast as possible. The reported latency (stride width) represents the delay introduced by model inference with a 2-second processing window. As a result, we confirmed that 2 seconds of audio data can be processed within an acceptable timeframe. Even in environments that do not support GPUs, processing can be completed within 120 milliseconds, making it effective for systems that require fast responses, such as spoken dialogue systems. Future work should include comprehensive comparisons with other real-time ASR systems and streaming implementations of whisper [23], [24], under identical conditions.

V. DISCUSSION

We analyzed the difference in performance between the 'w/o tag' and 'tagging' models when recognizing utterances containing non-speech segments, examining how this difference varied depending on the pre-trained ASR model employed. When the rinna wav2vec 2.0 model was used, the 'tagging' model performed 6% better at SNR = 0 dB, but 12% worse at SNR = 20 dB. When using the facebook wav2vec 2.0 model, however, the 'tagging' model consistently improved performance across all environments, showing up to 6% better recognition performance than the 'w/o tag' model. Similar performance improvements of up to 9% were also observed when using 'tagging' with the English ASR model, suggesting that tagging consistently facilitates acoustic discrimination between speech and non-speech segments when using the facebook model.

This difference in performance likely stems from the characteristics of the pre-training data: The Facebook wav2vec 2.0 model was pre-trained on the clean LibriSpeech corpus, whereas the Rinna model was pre-trained on the noisy ReazonSpeech data (e.g., including live TV sports broadcasts). During pre-training, wav2vec 2.0 acquires quantized acoustic features of 25 ms in width using contrastive loss, including from various non-speech segments. When fine-tuned with CTC, the model's blank tokens can implicitly act as non-speech indicators, serving a purpose similar to our proposed explicit non-speech tags. This explains why the 'w/o tag' model performed better with the Rinna model under certain conditions, as this might have avoided conflicts with the model's blank token representations, particularly in noisy environments (SNR = 5 dB and 0 dB).

TABLE V
PERFORMANCE COMPARISON OF JAPANESE ASR MODELS: INFERENCE SPEED AND ONLINE PROCESSING LATENCY WITH 2-SECOND WINDOW.

Model		RTF	latency
whisper-base		0.0191	N/A
whisper-large-v3-turbo		0.0220	
wav2vec 2.0 + VAD	text_c	0.0054	N/A
	seg_c	0.0065	
wav2vec 2.0	(GPU)	**0.0023**	15ms
	(M1)	0.0756	120ms

VI. CONCLUSION

We developed a real-time VAD-less speech recognition model by fine-tuning an SSL model on combined speech and non-speech audio signals, with non-speech sections explicitly tagged in the training data. The results of our experiments demonstrated that our proposed method achieved superior results compared to conventional VAD-based approaches when processing input data that includes non-speech segments. In addition, we found that the effectiveness of the assigned non-speech tags varied depending on the characteristics of the speech data used for pre-training the base model. In this study, we constructed the proposed ASR model based on existing pre-trained SSL models, but in the future, we will investigate the observations obtained by using models trained from scratch, such as Conformer [25], [26]. We also plan to apply the obtained VAD-less speech recognition model to streaming and take measures to mitigate performance degradation caused by the use of streaming data [27], [28].

VII. ACKNOWLEDGMENT

This research was supported by JSPS KAKENHI grants JP22K19793 and JP23H00493.

REFERENCES

[1] A. Baevski, H. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations," 2020. [Online]. Available: https://arxiv.org/abs/2006.11477

[2] W.-N. Hsu, B. Bolte, Y.-H. H. Tsai, K. Lakhotia, R. Salakhutdinov, and A. Mohamed, "HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units," 2021. [Online]. Available: https://arxiv.org/abs/2106.07447

[3] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, and V. Stoyanov, "Unsupervised Cross-lingual Representation Learning at Scale," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, D. Jurafsky, J. Chai, N. Schluter, and J. Tetreault, Eds. Online: Association for Computational Linguistics, Jul. 2020, pp. 8440–8451. [Online]. Available: https://aclanthology.org/2020.acl-main.747/

[4] A. Babu, C. Wang, A. Tjandra, K. Lakhotia, Q. Xu, N. Goyal, K. Singh, P. von Platen, Y. Saraf, J. Pino, A. Baevski, A. Conneau, and M. Auli, "XLS-R: Self-supervised Cross-lingual Speech Representation Learning at Scale," 2021. [Online]. Available: https://arxiv.org/abs/2111.09296

[5] K. D. N, P. Wang, and B. Bozza, "Using Large Self-Supervised Models for Low-Resource Speech Recognition," in *Interspeech 2021*, 2021, pp. 2436–2440.

[6] M. Wiesner, D. Raj, and S. Khudanpur, "Injecting Text and Cross-lingual Supervision in Few-shot Learning from Self-Supervised Models," 2021. [Online]. Available: https://arxiv.org/abs/2110.04863

[7] A. Sriram, M. Auli, and A. Baevski, "Wav2Vec-Aug: Improved self-supervised training with limited data," 2022. [Online]. Available: https://arxiv.org/abs/2206.13654

[8] Q.-S. Zhu, J. Zhang, Z.-Q. Zhang, M.-H. Wu, X. Fang, and L.-R. Dai, "A Noise-Robust Self-Supervised Pre-Training Model Based Speech Representation Learning for Automatic Speech Recognition," in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, May 2022, p. 3174–3178. [Online]. Available: http://dx.doi.org/10.1109/ICASSP43922.2022.9747379

[9] T. Yoshimura, T. Hayashi, K. Takeda, and S. Watanabe, "End-to-End Automatic Speech Recognition Integrated With CTC-Based Voice Activity Detection," 2020. [Online]. Available: https://arxiv.org/abs/2002.00551

[10] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 369–376.

[11] A. Graves and N. Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in *International conference on machine learning*. PMLR, 2014, pp. 1764–1772.

[12] S. Novitasari, T. Fukuda, and G. Kurata, "Improving ASR Robustness in Noisy Condition Through VAD Integration," in *Interspeech 2022*, 2022, pp. 3784–3788.

[13] Panayotov, Vassil and Chen, Guoguo and Povey, Daniel and Khudanpur, Sanjeev, "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.

[14] Y. Yin, D. Mori, and S. Fujimoto, "ReazonSpeech: A Free and Massive Corpus for Japanese ASR," 2016.

[15] V. Andrew and H. J. Steeneken, "Assessment for automatic speech recognition: II. NOISEX-92: A database and an experiment to study the effect of additive noise on speech recognition systems," *Speech Communication*, vol. 12, no. 3, pp. 247–251, 07 1993. [Online]. Available: https://cir.nii.ac.jp/crid/1362262943981296384

[16] T. Kudo, "Subword Regularization: Improving Neural Network Translation Models with Multiple Subword Candidates," 2018. [Online]. Available: https://arxiv.org/abs/1804.10959

[17] T. Kudo and J. Richardson, "SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing," 2018. [Online]. Available: https://arxiv.org/abs/1808.06226

[18] K. Itou, M. Yamamoto, K. Takeda, T. Takezawa, T. Matsuoka, T. Kobayashi, K. Shikano, and S. Itahashi, "JNAS: Japanese speech corpus for large vocabulary continuous speech recognition research," *Journal of the Acoustical Society of Japan (E)*, vol. 20, no. 3, pp. 199–206, 1999.

[19] R. Ardila, M. Branson, K. Davis, M. Henretty, M. Kohler, J. Meyer, R. Morais, L. Saunders, F. M. Tyers, and G. Weber, "Common Voice: A Massively-Multilingual Speech Corpus," 2020. [Online]. Available: https://arxiv.org/abs/1912.06670

[20] H. Bredin and A. Laurent, "End-to-end speaker segmentation for overlap-aware resegmentation," in *Proc. Interspeech 2021*, Brno, Czech Republic, August 2021.

[21] H. Bredin, R. Yin, J. M. Coria, G. Gelly, P. Korshunov, M. Lavechin, D. Fustes, H. Titeux, W. Bouaziz, and M.-P. Gill, "pyannote.audio: neural building blocks for speaker diarization," in *ICASSP 2020, IEEE International Conference on Acoustics, Speech, and Signal Processing*, Barcelona, Spain, May 2020.

[22] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust Speech Recognition via Large-Scale Weak Supervision," 2022. [Online]. Available: https://arxiv.org/abs/2212.04356

[23] D. Macháček, R. Dabre, and O. Bojar, "Turning whisper into real-time transcription system," in *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics: System Demonstrations*, S. Saha and H. Sujaini, Eds. Bali, Indonesia: Association for Computational Linguistics, Nov. 2023, pp. 17–24. [Online]. Available: https://aclanthology.org/2023.ijcnlpdemo.3

[24] H. Wang, G. Hu, G. Lin, W.-Q. Zhang, and J. Li, "Simul-whisper: Attention-guided streaming whisper with truncation detection," in *Interspeech 2024*, 2024, pp. 4483–4487.

[25] 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," 2020. [Online]. Available: https://arxiv.org/abs/2005.08100

[26] D. Rekesh, N. R. Koluguri, S. Kriman, S. Majumdar, V. Noroozi, H. Huang, O. Hrinchuk, K. Puvvada, A. Kumar, J. Balam, and B. Ginsburg, "Fast conformer with linearly scalable attention for efficient speech recognition," 2023. [Online]. Available: https://arxiv.org/abs/2305.05084

[27] Doutre, Thibault and Han, Wei and Ma, Min and Lu, Zhiyun and Chiu, Chung-Cheng and Pang, Ruoming and Narayanan, Arun and Misra, Ananya and Zhang, Yu and Cao, Liangliang, "Improving streaming automatic speech recognition with non-streaming model distillation on unsupervised data," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 6558–6562.

[28] T. Doutre, W. Han, C.-C. Chiu, R. Pang, O. Siohan, and L. Cao, "Bridging the gap between streaming and non-streaming ASR systems bydistilling ensembles of CTC and RNN-T models," 2021. [Online]. Available: https://arxiv.org/abs/2104.14346