Interspeech 2024
1-5 September 2024, Kos, Greece

VAE-based Phoneme Alignment Using Gradient Annealing and SSL Acoustic Features

Tomoki Koriyama[1]

[1]CyberAgent, Japan

koriyama_tomoki@cyberagent.co.jp

Abstract

This paper presents an accurate phoneme alignment model that aims for speech analysis and video content creation. We propose a variational autoencoder (VAE)-based alignment model in which a probable path is searched using encoded acoustic and linguistic embeddings in an unsupervised manner. Our proposed model is based on one TTS alignment (OTA) and extended to obtain phoneme boundaries. Specifically, we incorporate a VAE architecture to maintain consistency between the embedding and input, apply gradient annealing to avoid local optimum during training, and introduce a self-supervised learning (SSL)-based acoustic-feature input and state-level linguistic unit to utilize rich and detailed information. Experimental results show that the proposed model generated phoneme boundaries closer to annotated ones compared with the conventional OTA model, the CTC-based segmentation model, and the widely-used tool MFA.

Index Terms: phoneme alignment, annealing, self-supervised learning, forward-sum algorithm, variational autoencoder

1. Introduction

Alignment in speech processing refers to the task of determining the times at which phonemes, characters, or words are uttered, based on a pair of a speech waveform and a phoneme or character sequence. Accurate alignment can be used to analyze phonetic and prosodic features of speech in detail, as well as to create video content such as subtitles, lip-sync, and speech editing. Furthermore, the alignment between the speech signal and the phoneme or text is crucial for modeling speech recognition and synthesis, because these tasks involve sequence-to-sequence transformations with different lengths.

The most well-known conventional method of phoneme alignment is the use of the Gaussian mixture model (GMM)-based hidden Markov model (HMMs) [1], which is also used the recent tool, Montreal Forced Aligner (MFA) [2]. To overcome the model expressiveness of GMM, many neural network (NN)-based methods have been proposed [3–14], from the perspective of automatic speech recognition (ASR) and text-to-speech (TTS). CTC segmentation [3] utilizes an ASR network trained with connectionist temporal classification (CTC) loss that includes a blank character. Since CTC segmentation only requires ASR models, it is easy to adapt various languages as WhisperX does [4]. However, the blank symbol conceals the beginning and end of each character, resulting in inaccurate boundary prediction. Another approach uses supervised training to predict the boundaries of phonemes [5]. The issue with using a supervised method is that it is applicable to various languages or speaking styles because phoneme boundary annotation for training is very limited.

TTS research also developed alignment model to capture the relationship between text and waveform. Famous frameworks use a forward-sum algorithm [11, 12] or its Viterbi search variant, known as monotonic alignment search (MAS) [13], which searchs for a monotonic path with dynamic programming. For example, Glow-TTS [13] and VITS [14] perform MAS using acoustic embeddings extracted by normalizing flow architecture, while RAD-TTS [12] and one tts alignment (OTA) [11] utilize forward-sum loss with NN-based linguistic and acoustic encoders. The use of forward-sum loss enables simultaneous training of alignment and TTS. However, the phoneme boundaries obtained from alignment modules are not always accurate, even though the synthetic speech is natural sounding due to the high expressiveness of NN models. Moreover, the accuracy of alignment has not been precisely evaluated in the previous studies [11, 14].

In this study, we explore phoneme alignment that can predict accurate boundaries in an unsupervised manner. Specifically, we propose a novel alignment model by extending the alignment module of OTA [11]. We incorporate the variational autoencoder (VAE) [15] architecture to retain time-dependent acoustic and linguistic information of encoders. We also introduce acoustic features obtained from self-supervised learning (SSL) models [16, 17] to utilize rich acoustic information, similar to recent alignment studies [4, 8]. Furthermore, we investigate the use of gradient annealing to avoid local optima during the path search, inspired by the gradient-based GMM training technique [18].

Experimental evaluations are conducted using a corpus that include manually-annotated phoneme boundaries. We calculate the errors of boundaries predicted by various modeling settings and demonstrate that the proposed method can yield more accurate phoneme boundaries compared with the conventional methods. Moreover, we perform an ablation study to examine the effectiveness of each component of the proposed model. Some examples are available at [1].

2. One TTS Alignment

A popular method for text or phoneme alignment involves searching for an appropriate path between an acoustic embedding sequence $(\mathbf{y}_1, \ldots, \mathbf{y}_T)$ and a linguistic feature sequence $(\mathbf{x}_1, \ldots, \mathbf{x}_K)$. The parallel version of OTA [11], which we refer to as OTA in this paper and is derived from RAD-TTS [12], uses convolutional networks to obtain the embeddings and trains the network by minimizing the following forward-sum

[1] https://github.com/hyama5/vae_align

10.21437/Interspeech.2024-1127

loss function:

$$L_{\text{align}} = -\log \sum_{\mathbf{s}\in S} \prod_{t=1}^{T} b(t, s_t). \quad (1)$$

Here, $\mathbf{s} = (s_1, \ldots, s_T)$ $(s_t \in \{1, \ldots, K\})$ is the state sequence and S is a set of monotonic left-to-right no-skip paths. The function $b(t, s_t) > 0$ represents the likelihood that the acoustic feature at time t has the s_t-th linguistic feature, which is defined by the product of a matching function $f(\mathbf{y}_t, \mathbf{x}_k)$ and a position prior $p_{\text{pos}}(t, k; \omega)$ as follows:

$$b(t, k) = f(\mathbf{y}_t, \mathbf{x}_k) p_{\text{pos}}(t, k; \omega). \quad (2)$$

The matching function $f(\cdot)$ is a softmax function with the logits of negative squared distance given by

$$f(\mathbf{y}_t, \mathbf{x}_k) = \frac{\exp(-\|\mathbf{y}_t - \mathbf{x}_k\|^2)}{\sum_{i=1}^{K} \exp(-\|\mathbf{y}_t - \mathbf{x}_i\|^2)}. \quad (3)$$

The position prior $p_{\text{pos}}(\cdot)$ is a bias function defined independently from acoustic and linguistic features to represent the rough diagonal tendency of paths. Specifically, the position prior is implemented using a beta-binomial distribution $p_{\text{pos}}(t, k; \omega) = \text{BetaBin}(k, \omega t, \omega(T - t + 1))$. Although a large value of ω helps fast convergence during training, it can also impose a strong restriction because not all speech utterances have diagonal paths.

The training of encoders and decoders is performed by gradient-based optimization using the forward-sum loss L_{align}. The gradient for $\log b(\cdot)$ is given by

$$\frac{\partial L_{\text{align}}}{\partial \log b(t, k)} = -\frac{\sum_{\mathbf{s}\in S(t,k)} \prod_{\tau=1}^{T} b(\tau, s_\tau)}{\sum_{\mathbf{s}\in S} \prod_{\tau=1}^{T} b(\tau, s_\tau)} \triangleq -\gamma(t, k) \quad (4)$$

where $S(t, k)$ is the set of paths through state k at time t. The term $\gamma(t, k)$ represents the probability of passing through (t, k), which is known as state occupancy in the training of HMM. It is possible to compute the $\gamma(t, k)$ using the dynamic programming of the forward-backward algorithm in a similar manner to HMM. When predicting phoneme boundaries from the trained model, we perform a Viterbi search, also known as monotonic align search, to find the most likely path and identify the state for each time.

3. Proposed VAE-based Alignment

We propose a novel alignment model by extending OTA. The major modifications from OTA include a VAE-based architecture, gradient annealing, SSL acoustic features, and state-level linguistic units.

For accurate alignment, the acoustic embedding of the t-th frame should retain the characteristics of the acoustic feature of the t-th frame, and the same applies to linguistic embeddings. However, we have observed that when we use the OTA model, the acoustic and linguistic embeddings do not always maintain the original acoustic and linguistic information and often overfit the alignment loss. To address this issue, we incorporate a VAE architecture [15] as shown in Fig. 1. We expect that this architecture will preserve the input acoustic and linguistic information in the embedding layers. The loss function is defined by

$$L = L_{\text{align}} + w_{\text{aco}} L_{\text{aco}} + w_{\text{lng}} L_{\text{lng}} \quad (5)$$

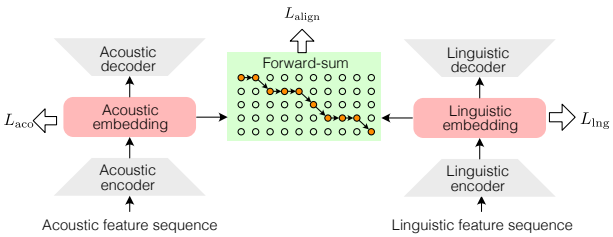


Figure 1: *Architecture of the proposed alignment model.*

where w_{aco} and w_{lng} are weights, and L_{aco} and L_{lng} are VAE losses, respectively.

Moreover, gradient-based training often falls into a local optimum. That is, the path becomes fixed in the early stages of training, making it difficult to search for the probable path during training. This issue aises from the gradient given in Eq. (4). If the likelihood $b(t, k)$ is very small, the gradient becomes close to zero, and thus the parameters for the likelihood $b(t, k)$ are slightly updated. This phenomenon is observed in similar models such as HMM or GMM that have hidden states when we perform gradient-based optimization [18]. Therefore, we propose annealing for the gradients in the alignment, based on the annealing in GMM training [18]. Specifically, we employ convolution with a Gaussian annealing filter $g(\cdot)$ given by

$$\frac{\partial L_{\text{align}}}{\partial \log b(t, k)} \approx -\gamma'(t, k) = -\gamma(t, k) * g(k) \quad (6)$$

$$g(k) = \exp\left(-\frac{k^2}{2\sigma^2}\right) \quad (7)$$

where σ is the annealing coefficient, and $*$ denotes convolution. The coefficient σ is updated by multiplying a rate $r < 1.0$ during training. The annealed gradient approximates the original one in Eq. (4) when σ becomes sufficiently small. This annealing shares the possibility with adjacent states, which helps to avoid the local optimum problem.

The use of SSL models has had a significant impact on the field of speech processing. For example in alignment, the CTC segmentation demo [19] and WhisperX [4] utilize SSL models to perform alignment. We apply SSL features extracted from a pretrained SSL model to the input of acoustic features. Since SSL models are trained by predicting the masked frame-level components, the SSL features can provide position-dependent rich information that is useful for accurate alignment.

We also examine the alignment level of linguistic units. In OTA [11] and other models [3, 12, 14], phonemes or characters are used as the unit of alignment. This assumes that each phoneme or character has one stable feature in acoustic embeddings. However, some phonemes change drastically within themselves. For example, a plosive sound has two different segments: a silent part and a bursting part, which is often referred to as subphone. Therefore, we extend the linguistic unit to the multiple-state level by dividing the linguistic input of one phoneme into multiple states per phoneme. This approach allows us to capture the changes within the phonemes, similar to the GMM-HMM-based methods.

4. Experiments

4.1. Experimental conditions

4.1.1. Dataset

We used the Corpus of Spontaneous Japanese (CSJ) [20] for experimental evaluations. The core data of CSJ, which includes detailed phoneme annotation, was divided into training, development, and evaluation subsets based on the ASR recipe of ESPnet [21]. The training, development, and evaluation subsets comprised 24,863 utterances (188 talks, 1,580 k phonemes, 33.2 h), 900 utterances (8 talks, 56 k phonemes, 1.2 h), 896 utterances (8 talks, 62 k phonemes, 1.3 h), respectively. We used 52 entities as a phoneme set and regarded the annotated phoneme boundaries as the true ones for evaluation. It should be noted that these annotated boundaries were not used during training.

4.1.2. Acoustic features

We used mel-frequency cepstral coefficient (**MFCC**) and melspectrogram (**Melspec**) as conventional acoustic features. MFCC was a 39-dimensional feature consisting of 13-dimensional coefficients and their Δ and Δ^2 features. The dimension of melspectrogram was set to 80. The acoustic features were extracted with a 10 ms frame-shift. For SSL features, we used four types of SSL-based models trained with different objectives: **HuBERT(en)**, **HuBERT(ja)**, **X-LSR**, and **w2v2-ASR**. HuBERT(en) was a HuBERT-base model [17] trained with English Librispeech [22] database, whereas HuBERT(ja) was a HuBERT-base model trained with the Japanese ReazonSpeech [23] database [24], which is the same language as the target one. XLS-R was a wav2vec2.0 large model [25] trained with multiple languages. The model w2v2-ASR was fine-tuned from XLS-R using Japanese corpora. We also used Whisper encoder [26] (**Whisper**), which has recently gained popularity as a base model for feature extraction. Since SSL and Whisper features were defined with a 20 ms shift, we upsampled them to a 10 ms shift to compare with the conventional features.

4.1.3. Base model configurations

We define a base proposed model used for the following evaluations. The encoders and decoders for acoustic and linguistic features were composed of 6-layer convolution networks, respectively. The kernel size of convolution layers was 3 and 256 hidden channels were employed. The dimension of embeddings for the matching function was 64. We adopted 3 states per phoneme as the modeling level of linguistic units. The loss weights were set as $(w_{\mathrm{aco}}, w_{\mathrm{lng}}) = (0.1, 0.1)$. The coefficient of position prior ω was set to 0.01. We set the initial annealing coefficient σ and update rate r to 30.0 and 0.9, respectively, and update σ every 1000 training steps. The training was performed up to 90k steps, and using Adam optimization [27] with a learning rate of 1.0×10^{-5}. The minibatch size was 4.

We implemented the computation of dynamic programming for the forward-sum loss using Triton [28] and other computations using PyTorch. Training time with one NVIDIA A100 GPU took approximately 150 min.

4.1.4. Conventional methods

We used MFA, CTC segmentation, and OTA for comparison. The MFA was trained using the same dataset as the other models utilizing the default configuration of MFA. To create the CTC-segmentation model, we trained a phoneme recognition model composed of a 6-layer Conformer [29] with CTC loss. As the input of the model, we used the output of HuBERT(ja) L12. The configuration of the OTA model was based on the original OTA paper [11]. Specifically, we used two and one convolution layers for acoustic and linguistic encoders, respectively. The acoustic features were melspectrogram, and the modeling level of linguistic features was a phoneme. In this study, only alignment loss was used instead of TTS loss for the OTA model. We used the same configurations as the proposed base model for other experimental settings.

Table 1: *Mean and median of boundary errors [ms] and tolerance error rates [%] on different acoustic features.*

Features	MAE	Median	20 ms tol	50 ms tol
MFCC	15.29	10.24	21.0	3.57
Melspec	15.46	10.59	21.8	3.20
HuBERT(en) L0	14.13	8.49	18.3	3.45
HuBERT(en) L4	13.33	8.67	18.0	2.45
HuBERT(en) L8	13.53	9.02	18.0	2.40
HuBERT(en) L12	13.53	9.03	18.4	2.40
HuBERT(ja) L0	**12.91**	**8.25**	**16.1**	2.59
HuBERT(ja) L4	13.40	8.69	17.8	2.56
HuBERT(ja) L8	14.05	10.46	19.8	**2.13**
HuBERT(ja) L12	13.40	9.33	17.7	2.21
XLS-R L0	13.66	8.50	17.4	3.16
XLS-R L8	13.68	8.50	17.4	3.16
XLS-R L16	13.66	8.50	17.3	3.17
XLS-R L24	13.79	8.92	17.9	2.75
w2v2-ASR L0	13.62	8.56	17.3	3.08
w2v2 ASR L8	14.46	9.93	21.0	2.83
w2v2-ASR L16	17.92	12.56	30.4	4.08
w2v2-ASR L24	17.33	12.05	28.6	4.58
Whisper	22.89	19.54	48.6	4.93

4.1.5. Evaluation metrics

We evaluated the performance of the alignment models by calculating the errors between the predicted and annotated phoneme boundaries. There is no consensus on the accuracy of phoneme boundaries, which depends on the granularity of words and phonemes to be analyzed. Hence, we used the four evaluation metrics. **MAE** and **Median** were the mean and median of absolute errors. MAE was more sensitive to outliers than Median. The other metrics were tolerance error rates defined as the ratio of errors whose absolute values exceeded a specific threshold. We used 20 ms and 50 ms as the threshold. All values in the evaluation, except for MFA, were the average of 5 trials with different random seeds.

4.2. Results on acoustic features

We first evaluated the effect of acoustic features on phoneme alignment. Table 1 shows the results for each feature. In the table, L0 to L24 represent the layer indices of intermediate transformer layers. When we compare the conventional features of MFCC and Melspec with HuBERT and XLS-R, HuBERT and XLS-R features had smaller MAEs. Since the differences among HuBERT(en), HuBERT(ja), and XLS-R were not significant, the dependency of language and SSL modeling method may not be crucial. It is observed that w2v2-ASR had large errors when we used layers L16 and L24. A possible reason is that the fine-tuned model for ASR lost the position information,

Table 2: *Mean and median of boundary errors [ms] and tolerance error rates [%] on different alignment methods.*

Method	MAE	Median	20 ms tol	50 ms tol
MFA	16.46	10.45	23.1	4.31
CTC	40.81	34.63	70.9	30.87
OTA	23.10	14.55	36.1	7.28
Proposed	12.91	8.25	16.1	2.59

Table 3: *Mean and median of boundary errors [ms] and tolerance error rates [%] on ablation of model components.*

Method	MAE	Median	20 ms tol	50 ms tol
Proposed	12.91	8.25	16.1	2.59
w/o VAE	14.96	10.41	23.1	2.90
w/o annealing	13.26	10.45	16.8	2.89
Phoneme-level	16.09	10.29	22.9	4.22
w/o position prior	12.96	8.24	16.1	2.61

leading to worse alignment accuracy. This is supported by the large errors of Whisper encoders trained for speech recognition objectives. The following experiments were performed using the best features, **HuBERT(ja) L0**.

4.3. Results on methods

We compared the proposed method with the conventional ones shown in Table 2. We see that CTC segmentation had larger errors than the other methods. This could be because the existence of blank symbols and the CTC loss did not assure the accurate position information. Although we tried multiple types of input features and encoder architectures, the same tendency was observed. We also confirmed that the proposed method, an extension of OTA model, reduced errors, indicating that the extension worked as expected. Although the GMM-HMM-based MFA yielded small errors, the proposed method outperformed MFA.

4.4. Ablation study on proposed components

The proposed method extends OTA models by adding multiple features. Hence, we performed an ablation study to evaluate the importance of each component.

- **W/o VAE** only used L_{align} for training instead of L.
- **W/o annealing** used the gradient of Eq. (4) directly for training.
- **Phoneme-level** employed phonemes as the linguistic unit instead of three states per phoneme.
- **W/o position prior** did not use position prior of Eq. (2). This ablation was an evaluation of the position prior used in the OTA model.

The results are shown in Table 3. It is seen that the proposed VAE, gradient annealing, and state-level modeling were all important to reduce errors. The position prior did not significantly affect the errors. We observed that while the position prior was an important factor in convergence, since annealing was also effective for convergence, the position prior was not always necessary.

Table 4: *Mean and median of boundary errors [ms] and tolerance error rates [%] on ablation of encoders/decoders.*

Method	MAE	Median	20 ms tol	50 ms tol
Proposed base	12.91	8.25	16.1	2.59
Conformer aco. enc.	12.93	8.31	16.0	2.65
Conformer aco. dec.	12.84	7.86	15.5	2.75
Conformer ling. enc.	**12.51**	**8.21**	**15.4**	**2.32**
Conformer ling. dec.	13.18	8.51	16.5	2.69
Flow aco. enc	14.36	9.52	20.1	3.26

4.5. Ablation study on encoder and decoders

The encoders and decoders of base models were simply composed of 6 convolution layers. However, we can consider using more complex and highly expressive components. In this ablation study, we replaced the convolution layers of each acoustic/linguistic encoder or decoder with a 6-layer Conformer block [29], which included 4-head self-attention and depth-wise convolution modules. Another ablation involved using normalizing flow for acoustic encoder instead of VAE as in VITS [14]. We used an affine coupling layer in the same way as VITS, and trained the model of 6 flow steps and a 1-layer convolution for each flow step.

The results are shown in Table 4. It is seen that the use of Conformer for the linguistic encoder reduced errors compared with the proposed base model. A possible reason is that long-range linguistic contexts obtained by the Conformer could effectively extract linguistic embeddings for alignment. The errors using the flow-based acoustic encoder were larger than the proposed base model. This could be because of the structural limitation that flows are required to be invertible.

5. Conclusions

In this study, we proposed an unsupervised phoneme alignment model based on the OTA model. Specifically, we incorporated a VAE architecture to maintain consistency between inputs and embeddings, applied gradient annealing to avoid local optima, and incorporated SSL acoustic feature input and state-level linguistic unit to utilize rich and detailed information. In experimental evaluations, we measured alignment accuracy using annotated phoneme boundaries and demonstrated that the proposed method can generate more accurate alignment than conventional methods, including GMM-HMM-based ones. In future work, we should evaluate the effectiveness of the proposed model using multiple languages including low-resource ones and diverse styles such as expressive speech and singing voices. We will also investigate the effectiveness of the accurate alignment for practical applications such as TTS and video content creation.

6. References

[1] L. Rabiner, M. Sambur, and C. Schmidt, "Applications of a nonlinear smoothing algorithm to speech processing," *IEEE Trans. Acous., Speech Signal Process.*, vol. 23, no. 6, pp. 552–557, 1975.

[2] M. McAuliffe, M. Socolof, S. Mihuc, M. Wagner, and M. Sonderegger, "Montreal forced aligner: Trainable text-speech alignment using kaldi." in *Proc. Interspeech*, vol. 2017, 2017, pp. 498–502.

[3] L. Kürzinger, D. Winkelbauer, L. Li, T. Watzel, and G. Rigoll, "CTC-segmentation of large corpora for german end-to-end

speech recognition," in *Proc. SPECOM*. Springer, 2020, pp. 267–278.

[4] M. Bain, J. Huh, T. Han, and A. Zisserman, "WhisperX: Time-accurate speech transcription of long-form audio," in *Proc. INTERSPEECH*, 2023, pp. 4489–4493.

[5] J. Li, Y. Meng, Z. Wu, H. Meng, Q. Tian, Y. Wang, and Y. Wang, "NeuFA: Neural network based end-to-end forced alignment with bidirectional attention mechanism," in *Proc. ICASSP*, 2022, pp. 8007–8011.

[6] H. Chung, S.-H. Lee, and S.-W. Lee, "Reinforce-Aligner: Reinforcement Alignment Search for Robust End-to-End Text-to-Speech," in *Proc. Interspeech*, 2021, pp. 3635–3639.

[7] M. C. Kelley and B. V. Tucker, "A Comparison of Input Types to a Deep Neural Network-based Forced Aligner," in *Proc. Interspeech*, 2018, pp. 1205–1209.

[8] J. Zhu, C. Zhang, and D. Jurgens, "Phone-to-audio alignment without text: A semi-supervised approach," in *Proc. ICASSP*, 2022, pp. 8167–8171.

[9] T. Kouzelis, G. Paraskevopoulos, A. Katsamanis, and V. Katsouros, "Weakly-supervised forced alignment of disfluent speech using phoneme-level modeling," in *Proc. INTERSPEECH*, 2023, pp. 1563–1567.

[10] T. Raissi, W. Zhou, S. Berger, R. Schlüter, and H. Ney, "HMM vs. CTC for automatic speech recognition: Comparison based on full-sum training from scratch," in *Proc. SLT*, 2023, pp. 287–294.

[11] R. Badlani, A. Łańcucki, K. J. Shih, R. Valle, W. Ping, and B. Catanzaro, "One TTS alignment to rule them all," in *Proc. ICASSP*, 2022, pp. 6092–6096.

[12] K. J. Shih, R. Valle, R. Badlani, A. Lancucki, W. Ping, and B. Catanzaro, "RAD-TTS: Parallel flow-based TTS with robust alignment learning and diverse synthesis," in *ICML Workshop on Invertible Neural Networks, Normalizing Flows, and Explicit Likelihood Models*, 2021.

[13] J. Kim, S. Kim, J. Kong, and S. Yoon, "Glow-TTS: A generative flow for text-to-speech via monotonic alignment search," in *Proc. NeurIPS*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33, 2020, pp. 8067–8077.

[14] J. Kim, J. Kong, and J. Son, "Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech," in *Proc. ICML*, 2021, pp. 5530–5540.

[15] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," in *Proc. ICLR*, 2014.

[16] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *Proc. NeurIPS*, vol. 33, pp. 12 449–12 460, 2020.

[17] 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," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 3451–3460, 2021.

[18] A. Gepperth and B. Pfülb, "Gradient-based training of Gaussian mixture models for high-dimensional streaming data," *Neural Processing Letters*, vol. 53, no. 6, pp. 4331–4348, 2021.

[19] FORCED ALIGNMENT WITH WAV2VEC2 https://pytorch.org/audio/stable/tutorials/forced_alignment_tutorial.html.

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

[21] S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N. Enrique Yalta Soplin, J. Heymann, M. Wiesner, N. Chen *et al.*, "ESPnet: End-to-end speech processing toolkit," in *Proc. Interspeech*, 2018.

[22] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an ASR corpus based on public domain audio books," in *Proc. ICASSP*, 2015, pp. 5206–5210.

[23] ReazonSpeech https://research.reazon.jp/projects/ReazonSpeech/.

[24] Rinna/japanese-hubert-base https://huggingface.co/rinna/japanese-hubert-base.

[25] A. Babu, C. Wang, A. Tjandra, K. Lakhotia, Q. Xu, N. Goyal, K. Singh, P. von Platen, Y. Saraf, J. Pino *et al.*, "XLS-R: Self-supervised cross-lingual speech representation learning at scale," *arXiv preprint arXiv:2111.09296*, 2021.

[26] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," in *Proc. ICML*, 2023, pp. 28 492–28 518.

[27] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. ICLR*, 2015.

[28] P. Tillet, H.-T. Kung, and D. Cox, "Triton: an intermediate language and compiler for tiled neural network computations," in *Proceedings of the 3rd ACM SIGPLAN International Workshop on Machine Learning and Programming Languages*, 2019, pp. 10–19.

[29] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu, and R. Pang, "Conformer: Convolution-augmented transformer for speech recognition," in *Proc. Interspeech*, 2020, pp. 5036–5040.