Interspeech 2025
17-21 August 2025, Rotterdam, The Netherlands

Multi-lingual and Zero-Shot Speech Recognition by Incorporating Classification of Language-Independent Articulatory Features

Ryo Magoshi[1], *Shinsuke Sakai*[1], *Jaeyoung Lee*[1], *Tatsuya Kawahara*[1]

[1]Graduate School of Informatics, Kyoto University, Japan
{magoshi, sakai, jaeyoung}@sap.ist.i.kyoto-u.ac.jp, kawahara@i.kyoto-u.ac.jp

Abstract

We address multi-lingual speech recognition including unknown or zero-shot languages based on the International Phonetic Alphabet (IPA) and articulatory features. Articulatory features are language-independent representations for IPA based on phonetic knowledge. In the previous studies, however, they were mostly limited to two dimensions of place of articulation and manner of articulation. Moreover, the classification of articulatory features were not well aligned with phone recognition. In this study, we adopt a comprehensive 24-dimensional vector representation, and propose a training method in which IPA tokens and their corresponding articulatory features are simultaneously predicted based on CTC alignment. Experiments are conducted by fine-tuning the wav2vec 2.0 XLS-R model over 22 languages, and the results demonstrated significant improvements on average as well as in zero-shot language settings.

Index Terms: zero-shot speech recognition, articulatory features, IPA, CTC

1. Introduction

In recent years, the advent of deep learning models has led to remarkable improvements in speech recognition [1]. However, achieving high performance in end-to-end models requires huge amounts of speech and transcription data. Consequently, speech recognition for low-resource languages, especially unseen or zero-shot languages, results in poor performance.

Speech recognition for very low-resource or endangered languages is of great significance, as it contributes to understanding of their linguistic structures and preservation of the culture associated with these languages. Since they often do not have orthographic systems, the IPA (International Phonetic Alphabet) is usually adopted in transcription of speech of those languages. One promising approach for automatic speech recognition of these languages is fine-tuning a large-scale model which is pre-trained over many languages, since it can utilize knowledge acquired during pre-training and can share IPA tokens over many languages. Moreover, incorporating phonetic knowledge can help improve the performance [2–4], but the previous studies have not sufficiently leveraged this knowledge for modeling and training of speech recognition.

To improve IPA-based speech recognition, we incorporate an explicit articulatory feature classification mechanism, which provides a language-independent phonetic representation. Articulatory feature representations have typically been deterministically described along two dimensions: place of articulation and manner of articulation. However, they cannot describe all of the IPA, and IPA has more elaborate and detailed description of articulatory features. We adopt a flexible representation with a 24 dimensional vector. This approach allows for a universal phonetic description that represents all of the IPA in a comprehensive manner.

In order to recognize both IPA tokens and corresponding articulatory features, synchronizing frame-wise articulatory feature classification with IPA token prediction is critical. However, this problem was not well addressed in the previous studies. In the proposed method, we first align IPA tokens at the frame level using Connectionist Temporal Classification (CTC) [5], then estimate corresponding articulatory features for each output frame. We introduce Articulatory Feature Classification Module (AFCM), and compute the cross-entropy loss between the expected articulatory features and the AFCM's output at the CTC-aligned frames. In this manner, the model training is conducted by synchronizing articulatory feature classification with IPA token prediction. We demonstrate that this method improves performance in speech recognition for multi-lingual and zero-shot scenarios.

Section 2 discusses previous studies on speech recognition using articulatory features. Section 3 then describes the classification mechanism of articulatory features and the proposed model architecture. Section 4 presents the experimental evaluations, followed by the conclusion in Section 5.

2. Related Work

2.1. Articulatory Features for IPA

Articulatory features are defined for IPA tokens, and are conventionally described on two dimensions: place of articulation and manner of articulation. However, this framework can only be used for basic consonants, and cannot represent sounds with diacritics, for example, aspirated sounds such as [p^{h}]. Mortensen et al. [6] redefined articulatory features and proposed Panphon, a model that converts IPA tokens into articulatory features. It defines 24 types of articulatory features, which can have three values: +1 (present), -1 (not present), or 0 (don't care). They have meanings such as syl (syllabic: whether it is the core of a syllable), son (sonorant: whether it is a resonant sound), and cons (consonantal: whether it is a consonant). By preprocessing with Panphon, each IPA token is converted into a 24-dimensional 3-value vector. This representation covers all consonants and vowels, including those which cannot be described by the conventional representation. Thus, this framework is more flexible, accurate, and comprehensive.

2.2. Modeling of Articulatory Features

Previous studies related to this study can be categorized into those utilizing articulatory features [7–10], those employing the IPA [11–18], and those incorporating both [19–22]. This section discusses the studies that are relevant to our work.

10.21437/Interspeech.2025-1827

Li et al. [20] applied a language-specific transformation matrix to map articulatory features to phonemes. This approach can be applied to zero-shot scenarios, but the transformation matrix was not shared across languages. Moreover, this approach adopted a simple cascading structure of connecting acoustic space, attribute space, and phoneme space, and trained it from scratch, resulting in poor performance.

Lee et al. [21] incorporated a transformation matrix from articulatory features to IPA tokens within their model. This implicitly learns the prediction of articulatory features, which assists IPA recognition. However, those articulatory features are not explicitly utilized as supervisory signals for model training.

Li et al. [10] concatenated articulatory features (manner and place of articulation) for end-to-end speech recognition. However, their approach relied on language-dependent orthographic representations, making it unsuitable for unseen languages. Additionally, their articulatory features were limited to a subset of places and manners of articulation.

Yen et al. [22] enhanced IPA prediction by incorporating recognition of manner and place of articulation as subtasks. They introduced three classifiers for IPA tokens, manner of articulation, and place of articulation in parallel. Predictions of manner and place of articulation were then transformed through projection matrices to assist IPA token recognition. Their approach employs multiple CTC loss functions for IPA token prediction alongside manner and place of articulation recognition. Since these loss functions are computed independently, IPA, manner, and place of articulation are not synchronously learned. This lack of alignment comes from a fundamental problem of the end-to-end speech recognition. Furthermore, similar to Li et al. [10], their approach used mutually exclusive categories of manner and place of articulation, which restricts the expressiveness of articulatory feature representation.

3. Proposed Method

In this study, we propose a language-independent IPA prediction model by defining the union of all IPA tokens across languages as the output vocabulary. Additionally, we define articulatory feature prediction as a subtask of CTC-based IPA prediction. The articulatory features are explicitly aligned with IPA token labels during training, and serve as augmented objectives synchronized with the output IPA tokens. Each IPA token is represented by 24 articulatory features, encoded with a 24-dimensional ternary vector, which provides richer information than two dimensions of manner and place of articulation. This approach enables the model to explicitly learn articulatory features in synchronization with IPA token prediction, constructing a language-independent speech recognition system.

3.1. IPA tokenization

The IPA labels are obtained by converting language-specific texts into IPA tokens using Epitran [23]. Since the model predicts IPA tokens as output, it must account for languages that do not use spaces to separate words. To maintain consistency, we omit spaces between words in IPA label data, designing a model to avoid predicting a space between words.

Diacritics (such as ʰ or ʲ) and coarticulation symbol ◌͡ are meaningful only when they are attached to other IPA characters. Taking these symbols into consideration, tokenization is defined such that each IPA unit corresponding to a single articulatory feature vector is treated as a single token. For example, [ʈ͡ʂʰifan] is tokenized as [‘ʈ͡ʂʰ’, ‘i’, ‘f’, ‘a’, ‘n’]. The union of all resulting tokens constitutes the output token set of the model.

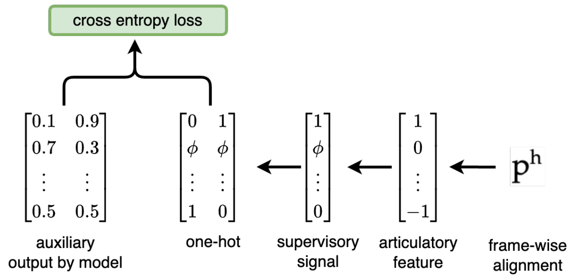

Figure 1: *Computation of the auxiliary loss. Non-contributory loss value is defined as ϕ, while -1 and 1 in articulatory features are converted to 0 and 1 for binary classification.*

3.2. Auxiliary Loss based on Articulatory Features

When a model being trained with CTC becomes relatively stable, it can compute an alignment between its output and the IPA labels. It allows for alignment of IPA tokens corresponding to each output frame, thereby obtaining the articulatory features for each frame. Note that, in CTC, there are many blank tokens, which do not result in any articulatory features. This means articulatory features are classified only at meaningful frames.

By converting the IPA tokens obtained through CTC alignment into 24-dimensional articulatory feature vectors, the ground-truth label for articulatory features is dynamically obtained. For non-IPA tokens (blank and padding tokens), a non-contributory loss value is assigned. Each dimension of the articulatory feature vector takes one of three values: 1 (the corresponding articulatory feature is present), -1 (absent), 0 (don’t care). Only dimensions with values of -1 or 1 contribute to the loss computation. This allows the auxiliary task to be formulated as a binary classification for each articulatory feature dimension. By designing the auxiliary output such that each frame produces a (24, 2) matrix, and given that the ground-truth articulatory feature vector for each frame is a 24-dimensional binary vector, cross-entropy loss is computed for each frame and each dimension. The computation of this auxiliary loss is illustrated in Figure 1. Note that the binary vector has the third value for not calculating the loss.

3.3. Articulatory Feature Classification Module (AFCM)

The Articulatory Feature Classification Module (AFCM) is illustrated in Figure 2. It consists of three components: linear transformation unit, gating unit, and articulatory feature extraction unit, from left to right. Since articulatory feature prediction is defined as a (24, 2) matrix, the extraction unit first transforms the input vector into a 24×2 = 48-dimensional space using a linear transformation, then reshapes it into a (24, 2) matrix, applying the Softmax function. This serves as the auxiliary output for articulatory feature prediction, where auxiliary loss is computed. The output is converted back to a 48-dimensional vector before passing through a linear transformation to match the required output dimensions. The other linear transformation unit operates in parallel with the extraction unit. To regularize the contributions of the extraction unit and the linear transformation unit, gating is introduced. The gating weights are constrained between 0 and 1 with a sum of 1 using a Sigmoid function. With this structure, AFCM produces both a main output with an arbitrary dimension and an auxiliary output in the form of a (24, 2) matrix.

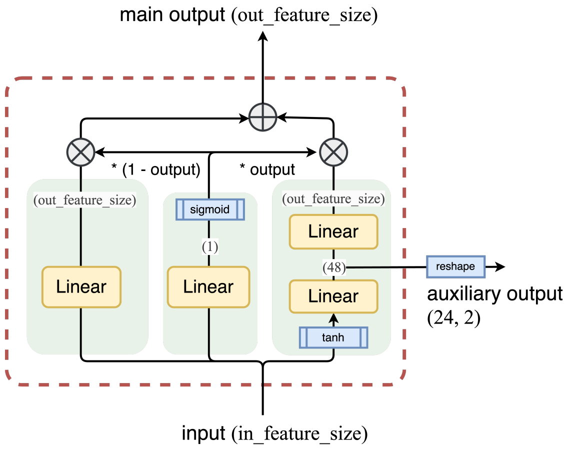

Figure 2: *Structure of Articulatory Feature Classification Module (AFCM). Auxiliary output is used for frame-wise classification of articulatory features.*

3.4. Entire Model Architecture

We adopt a pre-trained model of wav2vec 2.0 XLS-R [24, 25] for both of the baseline and the proposed method. The overall architecture of the proposed model is depicted in Figure 3. An AFCM is used as the final output layer instead of the conventional linear layer. Additionally, another AFCM is inserted between the Transformer layers of the wav2vec 2.0 encoder, and the main output undergoes GELU activation [26] and is residual-connected to the subsequent Transformer layer input.

The model produces three types of outputs: the main output of the AFCM at the final layer, the auxiliary output of the AFCM at the final layer, and the auxiliary output of AFCM between the intermediate layers. CTC loss ($\mathcal{L}_{CTC}$) is computed with the main output at the final layer and the ground-truth IPA label sequence. During this process, alignments are computed to obtain frame-level IPA token sequences including blank tokens. The aligned IPA tokens are converted into the articulatory feature representation, which serves as the ground truth for the frame-wise auxiliary outputs. The two auxiliary outputs are trained to classify the articulatory features for each frame using cross-entropy loss ($\mathcal{L}_{CE1}, \mathcal{L}_{CE2}$) for the final layer and the intermediate layer, respectively.

The total loss $\mathcal{L}$ is defined as follows:

$$\mathcal{L} = \mathcal{L}_{CTC} + \lambda_{CE1}\mathcal{L}_{CE1} + \lambda_{CE2}\mathcal{L}_{CE2}$$

λ_{CE1} and λ_{CE2} are hyper-parameters that control the weights of auxiliary losses.

4. Experimental Evaluations

4.1. Dataset

For the experiment, we use multi-lingual speech datasets that include zero-shot languages. Common Voice [27] 16.1 is an open-source speech dataset provided by Mozilla, containing over 30,000 hours of speech data in 120 languages. From this dataset, we selected 23 languages for which Epitran [23] supports grapheme-to-phoneme (G2P) conversion. A subset of these languages was further sampled to create a dataset totaling 280.3 hours. The selected languages are split into four categories in evaluation: 13 languages have training data of approximately 20 hours for each, and are treated as relatively high-resource languages. Three languages have training data within a range of 4.93 to 7.62 hours, and are classified as middle-resource languages. Furthermore, six languages have training data within a range of 0 to 1 hour, and are categorized as low-resource languages. Ligurian is not used either in pre-training or fine-tuning, so it is categorized as a zero-shot language.

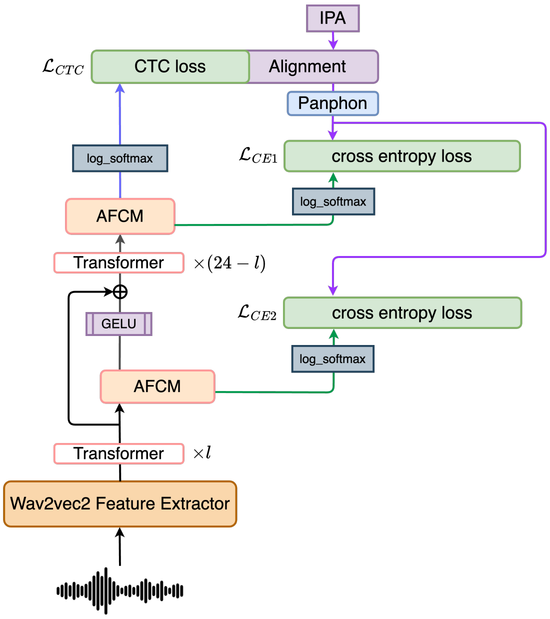

Figure 3: *Architecture of the proposed model. AFCMs are inserted between Transformer layers, and as the output layer.*

For another zero-shot language, we use Speech Corpus of Ainu Folklore [28, 29], which is based on oral recordings of folktales of the critically endangered Ainu language. In this experiment, we use the dataset described in [29], where eval1 serves as the test set and eval2 as the validation dataset. A single model is trained by using the two datasets to cover all of the languages.

4.2. Data Preprocessing

For Common Voice, G2P conversion is performed using Epitran for each language to obtain IPA sequences. There is no existing G2P model for Ainu, but the transcription of Ainu is primarily phonemic and written in the Latin alphabet, and most of the symbols are identical to IPA transcription. For a few exceptions, we employ a rule-based approach that maps tokens one-to-one to IPA sequences according to [30]. The text-to-IPA conversion rules are shown in Table 1. For context-dependent allophones due to regressive assimilation, IPA conversion is performed considering adjacent phonemes. Next, we tokenize the IPA sequences in the training data so that each single token is represented as a single articulatory feature vector. In these preprocessings, we obtained a vocabulary of 297 IPA tokens.

Table 1: *Conversion rules from Ainu orthography to IPA.*

Orthography	c	r	y	hi	hu	np	nk	=
IPA	t͡ʃ	ɾ	j	çi	Φu	mp	ŋk	

4.3. Experimental Settings

The backbone of both of the baseline and the proposed model is wav2vec 2.0 XLS-R 300m. The baseline model applies a linear

Table 2: *Speech recognition performance comparison by Transformer Layers to insert AFCM (CER [%]). Settings of high, mid, low, zero mean high-resource, middle-resource, low-resource, and zero-shot split, respectively. 5-6 means AFCM is inserted between the 5th and the 6th Transformer layers, in addition to the final layer, for example.*

Setting		high	mid	low	zero	ave
baseline		15.56	25.16	50.96	39.10	16.19
proposed	5-6	15.55	26.12	52.40	38.21	16.25
	7-8	14.94	24.69	53.48	38.38	15.66
	9-10	13.84	24.50	52.36	37.59	13.85
	11-12	14.99	24.87	49.97	37.96	14.99
	13-14	**12.40**	**24.40**	**46.86**	**36.48**	**13.12**
	15-16	14.41	26.49	51.75	36.86	14.42
	19-20	14.01	26.46	50.72	38.63	14.78
	23-24	14.18	26.53	49.89	39.35	14.93

layer to the output of the Transformer, and is trained using CTC loss. In the proposed model, the output linear layer is replaced with an AFCM, and another AFCM is inserted between some intermediate layers. The number of the parameters of AFCM is negligible compared to the baseline model.

Both of the baseline and the proposed model are trained with a batch size of 120. The learning rate is selected from $\{1 \times 10^{-4}, 2 \times 10^{-4}, 4 \times 10^{-4}, 8 \times 10^{-4}\}$ as a hyper-parameter, and is set to 2×10^{-4}. It increases linearly from 0 during the first 10% of the training steps and decreases linearly to 0 over the final 50%. The models are trained over 30 epochs with AdamW [31] optimizer. To mitigate the imbalanced language distribution, the probability of using each training sample is configured for its language. Using the temperature-based method [32], languages with less training data were set to be used more frequently than their original ratios. The temperature parameter τ is selected from $\{1, 4, 8\}$ as a hyper-parameter, and set to $\tau = 4$. All the hyper-parameters are selected so that the baseline performs the best. For evaluation, Character Error Rate (CER) is computed for the IPA characters. When calculating CER, diacritics as well as coarticulation symbols are treated as single characters.

In the proposed model, the Transformer layers to insert AFCM in-between is selected from the set {5-6, 7-8, 9-10, 11-12, 13-14, 15-16, 19-20, 23-24}, where 5-6 means AFCM is inserted between the 5th and the 6th Transformer layers, in addition to the final layer. The loss weights are set as $\lambda_{CE1} = 1$ and $\lambda_{CE2} = 1.5$.

4.4. Results

The results for each experimental setting are presented in Table 2. In all splits (high, mid, low, and zero), the proposed method achieves a better CER than the baseline. The setting of inserting the AFCM between the 13th and the 14th Transformer layers, which was identified as the optimal configuration for all splits, showed statistically significant improvements over the baseline at the 1% significance level for all splits. A relative improvement of 18.96% is gained on average.

Table 3 shows the comparison between the baseline and proposed method at its best setting. The proposed model outperforms the baseline in almost all of the languages. In the low-resource split, however, both of the baseline and proposed method exhibit low performance. This might to be caused not only by insufficient training data size, but also by G2P conversion errors for these languages. This problem needs to be addressed in the future.

Table 3: *Speech recognition performance comparison between the baseline and proposed method (CER [%]). The proposed method is on its best setting, where AFCM is inserted between the 13th and the 14th Transformer layers.*

Split	Language	train [hrs]	baseline	proposed
high	Kinyarwanda	20.5	23.74	**20.85**
	French	20.2	17.02	**13.92**
	German	20.2	14.00	**11.66**
	Swahili	20.1	16.64	**12.37**
	Chinese (China)	20.0	11.00	**10.74**
	Hungarian	20.0	9.70	**8.06**
	Turkish	20.0	20.71	**16.45**
	Polish	20.0	19.54	**11.88**
	Thai	20.0	16.90	**14.59**
	Italian	20.0	11.38	**7.50**
	Tamil	20.0	30.16	**22.53**
	Esperanto	19.8	5.98	**4.94**
	Spanish	19.8	11.15	**9.05**
mid	Indonesian	7.62	33.22	**31.92**
	Romanian	5.63	24.37	**22.89**
	Kurmanji Kurdish	4.93	**12.89**	14.68
low	Amharic	0.659	55.10	**53.43**
	Korean	0.618	48.26	**42.22**
	Lao	0.096	28.30	**26.42**
	Azerbaijani	0.070	58.21	**44.37**
	Telugu	0.044	**59.75**	61.68
	Tigrinya	0.021	76.92	**71.15**
zero	Ligurian	0	30.71	**29.89**
	Ainu	0	40.07	**37.23**

5. Conclusion

This study focuses on multi-lingual speech recognition whose output is represented by IPA tokens, so that the model can be utilized for transcription of zero-shot languages. The articulatory features of 24-dimensional vectors, which are adopted in this study, cover comprehensive phonetic knowledge in contrast to the previous studies. In the proposed model, the Articulatory Feature Classification Module (AFCM) identifies articulatory features, and frame-level synchronous articulatory feature labels are obtained from alignment using CTC. This allows for flexible and accurate training with improved supervisory signals. The proposed model incorporates the articulatory feature classification as an auxiliary task to enhance the token prediction instead of its use as pre-processing for the main task adopted in most of the previous works. This framework allows for effective yet robust enhancement, which is confirmed by consistent improvements over almost all languages including zero-shot languages. As a result, the improvements are much larger than those reported in the previous studies [20, 22], although the exact experimental settings are not the same.

6. Acknowledgements

This work was supported by JST NEXUS (JPMJNX25C1).

7. References

[1] J. Li *et al.*, "Recent advances in end-to-end automatic speech recognition," in *Proceedings of APSIPA Transactions on Signal and Information Processing*, vol. 11, no. 1. Now Publishers, Inc., 2022.

[2] S. M. Siniscalchi, D.-C. Lyu, T. Svendsen, and C.-H. Lee, "Experiments on cross-language attribute detection and phone recognition with minimal target-specific training data," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 3, pp. 875–887, 2011.

[3] D. Yu, S. M. Siniscalchi, L. Deng, and C.-H. Lee, "Boosting attribute and phone estimation accuracies with deep neural networks for detection-based speech recognition," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2012, pp. 4169–4172.

[4] C.-H. Lee and S. M. Siniscalchi, "An information-extraction approach to speech processing: Analysis, detection, verification, and recognition," *in Proceedings of the IEEE*, vol. 101, no. 5, pp. 1089–1115, 2013.

[5] 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.

[6] D. R. Mortensen, P. Littell, A. Bharadwaj, K. Goyal, C. Dyer, and L. Levin, "Panphon: A resource for mapping ipa segments to articulatory feature vectors," in *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, 2016, pp. 3475–3484.

[7] S. Stüker, F. Metze, T. Schultz, and A. Waibel, "Integrating multilingual articulatory features into speech recognition." in *Proceedings of Interspeech*, 2003, pp. 1033–1036.

[8] B. Abraham, S. Umesh, and N. M. Joy, "Articulatory feature extraction using ctc to build articulatory classifiers without forced frame alignments for speech recognition," in *Proceedings of Interspeech*, 2016, pp. 798–802.

[9] M. Müller, S. Stüker, and A. Waibel, "Towards improving low-resource speech recognition using articulatory and language features," in *Proceedings of the 13th International Conference on Spoken Language Translation*, M. Cettolo, J. Niehues, S. Stüker, L. Bentivogli, R. Cattoni, and M. Federico, Eds. Seattle, Washington D.C: International Workshop on Spoken Language Translation, Dec. 8-9 2016.

[10] S. Li, C. Ding, X. Lu, P. Shen, T. Kawahara, and H. Kawai, "End-to-end articulatory attribute modeling for low-resource multilingual speech recognition," in *Proceedings of Interspeech*, 2019, pp. 2145–2149.

[11] H. Lin, L. Deng, D. Yu, Y.-f. Gong, A. Acero, and C.-H. Lee, "A study on multilingual acoustic modeling for large vocabulary asr," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2009, pp. 4333–4336.

[12] X. Li, S. Dalmia, J. Li, M. Lee, P. Littell, J. Yao, A. Anastasopoulos, D. R. Mortensen, G. Neubig, A. W. Black *et al.*, "Universal phone recognition with a multilingual allophone system," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 8249–8253.

[13] Q. Xu, A. Baevski, and M. Auli, "Simple and effective zero-shot cross-lingual phoneme recognition," *arXiv preprint arXiv:2109.11680*, 2021.

[14] H. Wang, W. Zhang, H. Suo, and Y. Wan, "Multilingual zero resource speech recognition base on self-supervise pre-trained acoustic models," in *Proceedings of 13th International Symposium on Chinese Spoken Language Processing (ISCSLP)*, 2022, pp. 11–15.

[15] W. Lee, G. G. Lee, and Y. Kim, "Optimizing two-pass cross-lingual transfer learning: Phoneme recognition and phoneme to grapheme translation," in *Proceedings of IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2023, pp. 1–8.

[16] C. Taguchi, Y. Sakai, P. Haghani, and D. Chiang, "Universal automatic phonetic transcription into the international phonetic alphabet," in *Proceedings of Interspeech*, 2023, pp. 2548–2552.

[17] S. Feng, M. Tu, R. Xia, C. Huang, and Y. Wang, "Language-universal phonetic representation in multilingual speech pretraining for low-resource speech recognition," in *Proceedings of Interspeech*, 2023, pp. 1384–1388.

[18] P. C. English, E. A. Shams, J. D. Kelleher, and J. Carson-Berndsen, "Following the embedding: Identifying transition phenomena in wav2vec 2.0 representations of speech audio," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2024, pp. 6685–6689.

[19] J. Vásquez-Correa, P. Klumpp, J. R. Orozco-Arroyave, and E. Nöth, "Phonet: A tool based on gated recurrent neural networks to extract phonological posteriors from speech," in *Proceedings of Interspeech*, 2019, pp. 549–553.

[20] X. Li, S. Dalmia, D. Mortensen, J. Li, A. Black, and F. Metze, "Towards zero-shot learning for automatic phonemic transcription," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 05, Apr. 2020, pp. 8261–8268.

[21] J. Lee, M. Mimura, and T. Kawahara, "Embedding articulatory constraints for low-resource speech recognition based on large pre-trained model," in *Proceedings of Interspeech*, 2023.

[22] H. Yen, S. M. Siniscalchi, and C.-H. Lee, "Boosting end-to-end multilingual phoneme recognition through exploiting universal speech attributes constraints," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2024, pp. 11 876–11 880.

[23] D. R. Mortensen, S. Dalmia, and P. Littell, "Epitran: Precision g2p for many languages," in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC)*, 2018.

[24] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," in *Proceedings of neural information processing systems*, vol. 33, 2020, pp. 12 449–12 460.

[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] D. Hendrycks and K. Gimpel, "Gaussian error linear units (gelus)," *arXiv preprint arXiv:1606.08415*, 2016.

[27] 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," *arXiv preprint arXiv:1912.06670*, 2019.

[28] K. Matsuura, S. Ueno, M. Mimura, S. Sakai, and T. Kawahara, "Speech corpus of Ainu folklore and end-to-end speech recognition for Ainu language," in *Proceedings of the Twelfth Language Resources and Evaluation Conference*, N. Calzolari, F. Béchet, P. Blache, K. Choukri, C. Cieri, T. Declerck, S. Goggi, H. Isahara, B. Maegaard, J. Mariani, H. Mazo, A. Moreno, J. Odijk, and S. Piperidis, Eds. Marseille, France: European Language Resources Association, May 2020, pp. 2622–2628.

[29] T. Kawahara and K. Matsuura, "Diversity in languages and spoken language processing of ainu," *The Journal of the Acoustical Society of Japan*, vol. 81, no. 1, pp. 35–41, 2025.

[30] H. Nakagawa, *Ainu Go Koubunten (in Japanese)*. Hakusuisha, 2024, pp. 25–31.

[31] I. Loshchilov, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017.

[32] H. W. Chung, N. Constant, X. Garcia, A. Roberts, Y. Tay, S. Narang, and O. Firat, "Unimax: Fairer and more effective language sampling for large-scale multilingual pretraining," *arXiv preprint arXiv:2304.09151*, 2023.