Leveraging IPA and Articulatory Features as Effective Inductive Biases for Multilingual ASR Training

Jaeyoung Lee
Kyoto University
jaeyoung@sap.ist.i.kyoto-u.ac.jp

Masato Mimura
NTT Corporation
masato.mimura@ntt.com

Tatsuya Kawahara
Kyoto University
kawahara@i.kyoto-u.ac.jp

***Abstract*—In recent advancements in end-to-end ASR, large-scale self-supervised or weakly supervised models have achieved a significant milestone. However, it remains challenging to train consistently high-performing multilingual models, transferable to languages without much resource. In this study, we propose embedding universal phonological knowledge to multilingual ASR by predicting international phonetic alphabet (IPA) targets and universal articulatory features alongside primary grapheme targets. These additions are expected to provide effective inductive bias or regularization for predicting grapheme targets across various languages. In the experiments, which involve fine-tuning a pre-trained XLS-R model using 10,400 hours of data across 120 languages from the Common Voice corpus, our proposed method achieved a 6.81% relative reduction in character error rate.**

***Index Terms*—Multilingual ASR, Articulatory features, IPA**

I. Introduction

End-to-end Automatic Speech Recognition (ASR) has seen significant advancements with large-scale self-supervised and weakly supervised models. These models have also been shown effective for multilingual ASR [1] [2] [3], domain-shifted ASR [4], and low-resource scenarios [5].

Despite these successes, transferability between a large number of languages remains challenging. Specifically, simply increasing the number of languages does not guarantee positive transfer between languages, without considering language proximity or similarity [6]. Research has shown that using auxiliary objectives, such as language ID prediction [7], phoneme prediction or language-adversarial targets [6], can facilitate positive transfer in multilingual ASR. We focus on articulatory features in this study, along with international phonetic alphabet (IPA) labels, to improve multilingual ASR training by providing a language-universal representation.

Articulatory features describe the physical movements of speech organs during the production of speech sounds. These features are shared across different languages, even if they adopt different writing scripts or phoneme inventories [8]. Therefore, they can serve as a strong inductive bias in multilingual ASR. Recognition of articulatory features has been used for phone recognition [9], improving robustness to spontaneous and non-native speech [10], and enhancing performance in low-resource scenarios [11]. However, research leveraging articulatory features for general multilingual End-to-End ASR has been limited. While some efforts, such as [12], have incorporated articulatory feature predictions, they do not fully utilize the capabilities of large pre-trained speech models.

In this work, we propose a novel method to enhance end-to-end grapheme prediction models by incorporating auxiliary IPA targets and articulatory features. We use IPA symbols as an auxiliary output, which are language-independent, and enhance them with articulatory features - universal, low-dimensional phonological representations. Articulatory features are learned implicitly through a fixed mapping between articulatory features and IPA symbols, leveraging data from various phonologically diverse languages. The model is fine-tuned on the Common Voice 16.1 dataset using XLS-R, achieving a 6.81% relative reduction in character error rate (CER) compared to standard fine-tuning methods.

Our method is advantageous because it can be applied to any language in a straightforward manner. The method can also be extended to other frameworks such as RNN-Tranducer or sequence-to-sequence models, though our experiments are based on the CTC framework [13]. It is also scalable: given a grapheme-to-phoneme (G2P) tool, it can be extended to more languages with minimal effort, without requiring expert knowledge on the phonemes and manual annotation.

II. Related Work

A. Articulatory Modeling

Articulatory features are basic elements of speech and have been used in various ASR tasks. Li et al. [14] mapped input features into a low-dimensional *articulatory attribute space*, then used a frozen linear transformation (an attribute-to-IPA mapping) to convert this space into a phoneme target distribution. Training was performed using CTC loss on the phoneme target, allowing the articulatory space to be learned implicitly without explicit supervision. They demonstrated that the implicitly learned articulatory space improves zero-shot phoneme recognition performance.

Lee et al. [15] took a similar approach, adding a *free* layer to bypass the articulatory prediction layer to enhance IPA target prediction capabilities. They also *unfroze* the articulatory-to-IPA matrix for greater flexibility, showing its effectiveness in low-resource ASR with phonemes as the target.

Yen et al. [12] addressed multilingual ASR with auxiliary articulatory feature predictions, but similar to [15], they adopted phoneme target unit (rather than grapheme), which is a union of phonemes of the target languages. These approaches require phonological knowledge on the target languages.

Our approach is similar to that of [15], but we use graphemes as the primary target, which is more suitable for general multilingual ASR, with IPA and articulatory predictions serving as auxiliary targets. We significantly extend the scope by training on up to 120 languages, demonstrating the effectiveness of our method in massively multilingual settings. Also, instead of "unfreezing" the articulatory-to-IPA mapping, we employ a gating mechanism to address flexibility.

Because manually constructing the mapping between IPA and articulatory features is labor-intensive, we use PanPhon [16] to generate the mapping. PanPhon is a database for IPA symbols and their articulatory features. It defines 24 articulatory features for over 6,000 different IPA symbols, alleviating the need for manual embedding of phonetic/articulatory knowledge by human experts. The exact usage of this mapping will be explained in Section III-B.

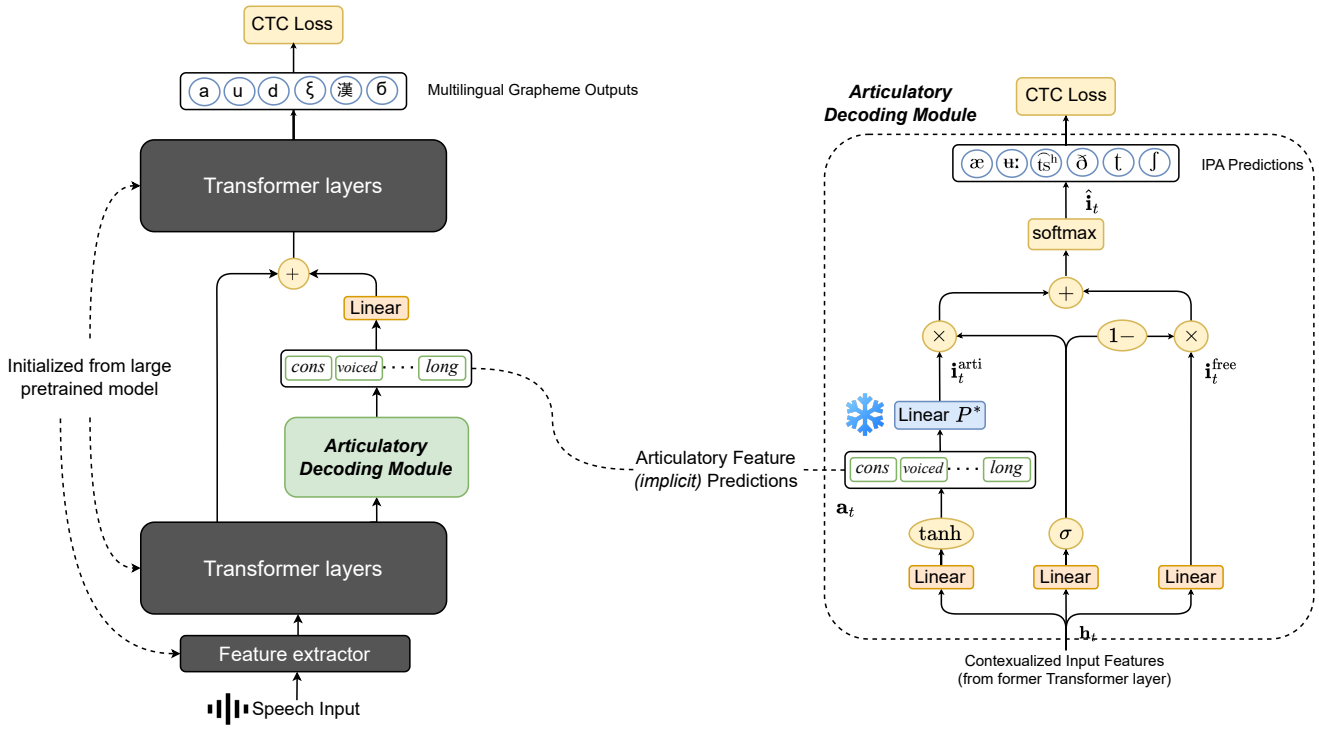

Fig. 1: An overview of the proposed method, applied to Transformer encoders and CTC loss in this case. Note that $\mathbf{a}_t$ denotes articulatory feature predictions, which are projected to IPA prediction space through the articulatory feature to IPA projection matrix P^*. Because the parameters of P^* are frozen, $\mathbf{a}_t$ can be indirectly trained to predict articulatory features without explicit supervision. In our experiments, the feature extractor and Transformer layers are initialized from pretrained XLS-R models.

B. Grapheme vs Phoneme Units

While grapheme units are widely used in end-to-end ASR as it is straightforward, many previous works mentioned in Section II-A used phoneme unit as the target. To that end, automatic grapheme-to-phoneme (G2P) conversion is conducted. Several G2P tools exist, such as Phonetisaurus [17], which uses the Weighted Finite-State Transducer (WFST) framework. In this study, we use a more recent model [18] based on ByT5 [19]. This model processes input and output tokens as bytes, supports up to 100 languages in a single model, and is trained end-to-end. We choose this model because it is easy to use, though it requires GPUs for operation, and expect that its byte-based nature will generalize better to unseen or rarely seen IPA symbols.

It should be noted, however, that like most G2P tools, it relies on word dictionaries and does not account for contextual phonetic variations caused by surrounding words (e.g., word-final consonants in French). The model's accuracy varies across languages: while about half of the 100 languages achieve less than 5% Phone Error Rate (PER), it can reach up to 30% for low-resource languages. This variability poses a major problem in adopting phoneme units as the primary target. Therefore, we use phoneme units as an auxiliary target.

III. Proposed Method

Our method is divided into two main components: multitask learning with IPA targets, and using articulatory feature predictions to infer IPA targets. Refer to Fig 1 for a schematic representation of our method.

A. Multitask Learning with IPA Symbols

In our method, we incorporate a secondary IPA target alongside the primary grapheme target for multilingual training. IPA pseudolabels are generated using a ByT5-based automatic G2P conversion tool, as described in [18]. These pseudolabels are not entirely accurate, but they assist in reinforcing the primary target training. Given an input speech signal X, a grapheme target Y, and an IPA target Z, the model is trained to minimize the following loss function:

$$\mathcal{L} = (1-\lambda)\log p(Y|X) + \lambda \log p(Z|X) \tag{1}$$

where λ denotes the weight of the IPA target loss. We achieve optimal performance when λ is linearly decreased according to the schedule:

$$\lambda_p = \lambda_0(1-p) \tag{2}$$

where p is the training phase from 0 to 1 and λ_0 is the initial value. This method allows the model to initially utilize the phonetic knowledge from the IPA targets and then gradually shift focus to the grapheme targets as training progresses.

TABLE I: IPA symbols and their articulatory features as defined in PanPhon [16]. The table shows a subset of the articulatory features; the full set includes 24 features, all of which are employed in our experiments.

	cons	*voiced*	*sg*	*high*	*long*	*tense*
/p/	+	-	-	-	-	0
/pʰ/	+	-	+	-	-	0
/pʲ/	+	-	-	+	-	0
/i/	-	+	-	+	-	+
/iː/	-	+	-	+	+	+

Although we use CTC loss in our experiments, our method can be easily applied to other frameworks, such as RNN-Transducer or sequence-to-sequence models, by placing the *Articulatory Decoding* module before the target unit output layer (refer to Fig. 1).

B. Articulatory Modeling

The IPA pseudolabels are designed to capture the universal phonetic representation of speech. However, they fall short of ideal due to the presence of language-specific IPA symbols. In contrast, articulatory features provide a universally shared basis across languages, with every IPA symbol definable in terms of these features. This capability facilitates the correlation of IPA symbols exclusive to certain languages, a relationship not encoded in standard end-to-end models. We adopt the articulatory features as defined in PanPhon [16], as demonstrated in Table I.

Each feature in PanPhon is encoded in a binary fashion; $+$ and $-$ denote the presence and absence of a feature, respectively, while 0 represents a *don't care* condition, indicating that the presence or absence of the feature is irrelevant to the specific IPA symbol. This ternary encoding focuses on phonological rather than phonetic contrasts, since accurate phonetic representation is inherently continuous, not binary or ternary. Consequently, precise phonetic labeling of IPA symbols becomes less critical, which is advantageous due to the error-prone nature of G2P conversion. Thus, automatic grapheme-to-phoneme conversion will be sufficient for our purpose.

C. Articulatory Feature Prediction

Here, we describe the module denoted as *Articulatory Decoding* in Fig. 1. Instead of directly predicting IPA targets, our model first infers articulatory features and then predicts IPA targets based on these inferred features. We define the *articulatory feature to IPA projection matrix* $P^* \in \mathbb{R}^{v \times f}$, where f denotes the number of articulatory features and v represents the number of IPA symbols. In this matrix, we encode the absence of a feature as -1, the presence as 1, and the *don't care* condition as 0. To normalize the influence of each feature, we divide each row of the matrix by the total number of non-zero features for that IPA symbol, ensuring the sum of the absolute values in each row equals 1. Thus, P^* acts as a soft lookup function.

Following the approach similar to [15], IPA targets are deduced by combining *free* outputs and *articulatorily constrained* outputs, represented as $\mathbf{i}_t^{\text{free}}$ and $\mathbf{i}_t^{\text{arti}}$, respectively. Given the encoded input features $\mathbf{h}_t$ at time t, the articulatory feature is calculated as follows:

$$\mathbf{a}_t = \tanh(\text{Linear}(\mathbf{h}_t)) \tag{3}$$

Subsequently, $\mathbf{i}_t^{\text{free}}$ and $\mathbf{i}_t^{\text{arti}}$ are determined as:

$$\mathbf{i}_t^{\text{free}} = \text{Linear}(\mathbf{h}_t) \tag{4}$$

$$\mathbf{i}_t^{\text{arti}} = P^* \mathbf{a}_t \tag{5}$$

The final IPA prediction, $\hat{\mathbf{i}}_t$, is derived by integrating $\mathbf{i}_t^{\text{free}}$ and $\mathbf{i}_t^{\text{arti}}$ through a gating mechanism:

$$\mathbf{w}_t = \sigma(\text{Linear}(\mathbf{h}_t)) \tag{6}$$

$$\hat{\mathbf{i}}_t = \text{Softmax}(\mathbf{w}_t \odot \mathbf{i}_t^{\text{arti}} + (1 - \mathbf{w}_t) \odot \mathbf{i}_t^{\text{free}}) \tag{7}$$

The implicitly predicted articulatory features, $\mathbf{a}_t$, get fed into the next transformer layer. Our approach diverges from [15] in that the parameters of P^* are fixed, enhancing the model's interpretability and flexibility through the incorporation of the gating mechanism.

TABLE II: Character error rates (CERs) for the *37-langs* setup and *all-langs* setup, across the three configurations: *baseline*, *ipa* and *articulatory*. The last column shows the relative error reductions (RERs) of the *articulatory* setup compared to the *baseline*. CERs are calculated for different groups of languages, categorized based on how much *train* data they have. Note that some languages were excluded from evaluation due to insufficient test data.

37-langs setup, using XLS-R(0.3B)

train size	# langs	*baseline*	*ipa*	*articulatory*	RER
<10h	5	12.33	11.33	**11.31**	8.33%
10-100h	16	7.67	6.87	**6.80**	11.29%
≥100h	10	4.95	4.58	**4.56**	8.00%
All	31	7.32	6.63	**6.58**	10.18%

all-langs setup, using XLS-R(1B)

train size	# langs	*baseline*	*ipa*	*articulatory*	RER
<10h	47	22.85	20.98	**20.56**	10.02%
10-100h	32	10.76	10.57	**10.30**	4.28%
≥100h	13	6.14	5.89	**5.63**	8.28%
All	92	10.43	10.00	**9.72**	6.81%

IV. Experimental Setup

A. Datasets

We used data from Common Voice V16.1 [20], which includes speech and text parallel data in 120 languages. Some languages in the original dataset had limited training data but more validation data. Therefore, we combined the train and a large portion of the dev set into a single *train* set, thus increasing the training data for tail languages. We formed the *dev* set by retaining only 20% of the original dev set, ensuring at least one hour of data remained for each language. If a language had less than an hour of data in the original Common Voice dataset, we retained the entire dev set for that language. We also ensured there is no overlap in `client_id` across the *train* and *dev* sets.

Our experiments involved two main setups: the *37-langs* setup, using a subset of 37 languages, and the *all-langs* setup, using all validated Common Voice data. For the *37-langs* setup, we selected languages that are well-supported by the ByT5-based G2P model [18], defined as having a phone error rate (PER) of 6.5% or lower. The *train* set included all 2,800 hours of data. In the final evaluation, we excluded 6 languages with less than 0.5 hours of data in the *test* set.

The *all-langs* setup included all languages and data of Common Voice 16.1, resulting in 10,400 hours of data for the *train* set. This setup included 13 languages with more than 100 hours of *train* data, 32 languages with 10-100 hours, and 43 languages with 1-10 hours. 28 languages were removed from the final evaluation as they contain less than 0.5 hours of *test* data.

B. Models

We employed the wav2vec 2.0-based large-scale and multilingual pretrained model, XLS-R [1], as the base model for our experiments. We used the 0.3B model for the *37-langs* setup and the 1B model for the *all-langs* setup. These models feature 24 and 48 transformer encoder layers, with embedding sizes of 1024 and 1280, respectively, and are trained on 436K hours of data across 128 different languages. Fine-tuning was conducted using CTC loss, with the convolution-based feature extractor frozen, while all encoder layers and output layers were fine-tuned using the Common Voice data.

TABLE III: Character error rate (CER) for various language categories with the XLS-R (1B) model across two configurations: *baseline* and *articulatory*. The last column shows the relative error reduction (RER) of the *articulatory* setup compared to the *baseline*.

Category	# languages	*train* (h)	*baseline*	*ipa*	*articulatory*	RER
Indo-Iranian	13	152	20.41	18.60	**18.04**	11.59%
Germanic	6	2761	7.02	6.87	**6.37**	9.34%
Slavic, Baltic	13	712	8.24	7.65	**7.52**	8.75%
Turkic	10	314	13.58	12.64	**12.53**	7.72%
Atlantic-Congo	5	1659	10.25	9.70	**9.50**	7.28%
Uralic	6	339	8.08	7.84	**7.52**	6.98%
Language Isolates	14	573	8.61	8.47	**8.02**	6.86%
Romance, Italic	12	3430	5.52	5.24	**5.21**	5.61%
Afro-Asiatic	4	204	18.60	18.94	**17.57**	5.54%
Dravidian	2	97	13.35	**12.14**	*13.52*	-1.25%
Sino-Tibetan, Japonic	7	109	**42.24**	*44.86*	*43.17*	-2.19%

We tested three model setups. In the *baseline* setup, a linear layer followed by a softmax was placed on top of the pretrained XLS-R to predict grapheme targets. In the *ipa* setup, the *Articulatory Decoding* module in Fig. 1 was replaced by a linear layer followed by a softmax, using $\mathbf{i}_t^{free}$ in the right-most path only, trained with IPA pseudolabels. The model then simply becomes multitask learning of grapheme and IPA targets, with the IPA output fed into the next Transformer layer. This setup is to verify the effectiveness of using articulatory features, on top of IPA. In the *articulatory* setup, articulatory feature prediction is employed as described in Section III and illustrated in Fig. 1. In all setups, the grapheme output used the final transformer layer (24th for the 0.3B model and 48th for the 1B model), while IPA and articulatory outputs used the output from the 3/4 top layer (18th for the 0.3B model and 36th for the 1B model).

The Adam optimizer was used with a learning rate that linearly warmed up for 10% of the steps, held constant for 40% of the steps, then linearly decayed to 0. For the 0.3B model, the batch size was set to 600 seconds with a peak learning rate of 2e-4 over 30,000 training steps. For the 1B model, the batch size was 800 seconds, with a learning rate of 5e-5 for over 50,000 training steps.

The vocabulary was shared across all languages in all setups, with no language-specific information provided during training or inference. The number of vocabulary (grapheme) is 1112 for 37 languages and 10592 for all (120) languages. In our experiments, the number of articulatory features is always 24, and the size of IPA vocabulary is 413 for experiments involving 37 languages and 688 for all (120) languages of Common Voice. The model was evaluated on *dev* set every 2% of training steps, and the best 5 models were averaged and evaluated on the *test* set. The character error rate (CER) from the main grapheme target output was used for evaluation.

V. Results

Table II shows the CER results for both the *37-langs* and *all-langs* setups. Our proposed method consistently improves performance across most languages, for both low-resource and high-resource languages. When comparing the *ipa* and *articulatory* setups, introducing IPA targets alone improves CER from 10.43 to 10.00 in the *all-langs* setup. Adding articulatory features in the *articulatory* setup further reduces the CER to 9.72, achieving a relative error reduction (RER) of 6.81%. For the *37-langs* setup, using the smaller XLS-R model (0.3B) resulted in a larger RER of 10.18%.

However, the impact of articulatory features in the *37-langs* setup is smaller compared to the *all-langs* setup. We suspect this is due to the smaller and less diverse IPA vocabulary in the *37-langs* setup. The larger IPA vocabulary in the *all-langs* setup likely benefits more from the articulatory features, as they capture phonological nuances more effectively across a broader range of languages.

A. Analysis of Results per Language Family

The CER results for various language categories with the XLS-R (1B) model across two configurations, *baseline* and *articulatory*, are presented in Table III. Languages are categorized based on their families, with single-language families (including constructed languages) classified as "language isolates".

Indo-Iranian languages showed the most significant improvement with the *articulatory* setup, achieving a relative error reduction (RER) of 11.59%. This substantial gain is likely due to the limited data for these languages and their diverse writing scripts. While diverse scripts typically challenge ASR performance, the proposed method mitigates this by leveraging universal articulatory features, which are less affected by script diversity.

Conversely, Sino-Tibetan and Japonic languages (including Chinese and Japanese) experienced a performance decline with the *ipa* and *articulatory* setups, showing a negative RER of -2.19%. Two main factors contribute to this: (1) the large target vocabulary size for these languages, ranging from 2500 to 5000 characters, and (2) poor G2P performance, with phone error rates between 10% and 30%. These factors likely overwhelm the benefits of the articulatory features, resulting in worse performance.

Overall, while the proposed method shows significant improvements in several language categories, particularly those with limited data and diverse scripts, it can be limited with languages with large vocabularies and poor G2P performance.

VI. Conclusion

Our study explored enhancing multilingual ASR through the integration of IPA and articulatory features, aiming for a universal representation of speech. We found that these features improve ASR performance, particularly in large-scale multilingual settings, reducing CER by relative 6.81% in experiments using 120 languages. Notably, our proposed method improves performance for both low- and high-resource languages. This is a promising direction for expanding multilingual ASR to massive number of languages.

VII. Acknowledgements

We would like to thank our colleague Kohei Matsuura from NTT Corporation for providing his technical expertise to help design the experiments.

REFERENCES

[1] Arun Babu, Changhan Wang, Andros Tjandra, Kushal Lakhotia, Qiantong Xu, Naman Goyal, Kritika Singh, Patrick von Platen, Yatharth Saraf, Juan Pino, Alexei Baevski, Alexis Conneau, and Michael Auli, "XLS-R: Self-supervised cross-lingual speech representation learning at scale," in *Proc. Interspeech 2022*, 2022, pp. 2278–2282.

[2] Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed, "HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units," June 2021.

[3] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever, "Robust Speech Recognition via Large-Scale Weak Supervision," 2022.

[4] Juan Zuluaga-Gomez, Amrutha Prasad, Iuliia Nigmatulina, Seyyed Saeed Sarfjoo, Petr Motlicek, Matthias Kleinert, Hartmut Helmke, Oliver Ohneiser, and Qingran Zhan, "How Does Pre-Trained Wav2Vec 2.0 Perform on Domain-Shifted Asr? an Extensive Benchmark on Air Traffic Control Communications," in *2022 IEEE Spoken Language Technology Workshop (SLT)*, Jan. 2023, pp. 205–212.

[5] Jing Zhao and Wei-Qiang Zhang, "Improving Automatic Speech Recognition Performance for Low-Resource Languages With Self-Supervised Models," *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, no. 6, pp. 1227–1241, Oct. 2022.

[6] Oliver Adams, Matthew Wiesner, Shinji Watanabe, and David Yarowsky, "Massively Multilingual Adversarial Speech Recognition," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Jill Burstein, Christy Doran, and Thamar Solorio, Eds., Minneapolis, Minnesota, June 2019, pp. 96–108, Association for Computational Linguistics.

[7] William Chen, Brian Yan, Jiatong Shi, Yifan Peng, Soumi Maiti, and Shinji Watanabe, "Improving Massively Multilingual ASR with Auxiliary CTC Objectives," in *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, June 2023, pp. 1–5.

[8] S. Stuker, T. Schultz, F. Metze, and A. Waibel, "Multilingual articulatory features," in *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03).*, Apr. 2003, vol. 1, pp. I–I.

[9] Dong Yu, Sabato Marco Siniscalchi, Li Deng, and Chin-Hui Lee, "Boosting attribute and phone estimation accuracies with deep neural networks for detection-based speech recognition," in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Kyoto, Japan, Mar. 2012, pp. 4169–4172, IEEE.

[10] Vikramjit Mitra, Wen Wang, Chris Bartels, Horacio Franco, and Dimitra Vergyri, "Articulatory Information and Multiview Features for Large Vocabulary Continuous Speech Recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Apr. 2018, pp. 5634–5638.

[11] Sheng Li, Chenchen Ding, Xugang Lu, Peng Shen, Tatsuya Kawahara, and Hisashi Kawai, "End-to-End Articulatory Attribute Modeling for Low-Resource Multilingual Speech Recognition," in *Interspeech 2019*. Sept. 2019, pp. 2145–2149, ISCA.

[12] Hao Yen, Sabato Marco Siniscalchi, and Chin-Hui Lee, "Boosting End-to-End Multilingual Phoneme Recognition Through Exploiting Universal Speech Attributes Constraints," in *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Apr. 2024, pp. 11876–11880.

[13] Alex Graves, Santiago Fernandez, Faustino Gomez, and Jurgen Schmidhuber, "Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks," 2006.

[14] Xinjian Li, Siddharth Dalmia, David Mortensen, Juncheng Li, Alan Black, and Florian Metze, "Towards Zero-Shot Learning for Automatic Phonemic Transcription," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 05, pp. 8261–8268, Apr. 2020.

[15] Jaeyoung Lee, Masato Mimura, and Tatsuya Kawahara, "Embedding Articulatory Constraints for Low-resource Speech Recognition Based on Large Pre-trained Model," in *INTERSPEECH 2023*. Aug. 2023, pp. 1394–1398, ISCA.

[16] David R. Mortensen, Patrick Littell, Akash Bharadwaj, Kartik Goyal, Chris Dyer, and Lori 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*, Osaka, Japan, Dec. 2016, pp. 3475–3484, The COLING 2016 Organizing Committee.

[17] Josef Robert Novak, Nobuaki Minematsu, and Keikichi Hirose, "Phonetisaurus: Exploring grapheme-to-phoneme conversion with joint n-gram models in the WFST framework," *Natural Language Engineering*, vol. 22, no. 6, pp. 907–938, Nov. 2016.

[18] Jian Zhu, Cong Zhang, and David Jurgens, "ByT5 model for massively multilingual grapheme-to-phoneme conversion," in *Proc. Interspeech 2022*, 2022, pp. 446–450.

[19] Linting Xue, Aditya Barua, Noah Constant, Rami Al-Rfou, Sharan Narang, Mihir Kale, Adam Roberts, and Colin Raffel, "ByT5: Towards a Token-Free Future with Pre-trained Byte-to-Byte Models," *Transactions of the Association for Computational Linguistics*, vol. 10, pp. 291–306, 2022.

[20] Rosana Ardila, Megan Branson, Kelly Davis, Michael Kohler, Josh Meyer, Michael Henretty, Reuben Morais, Lindsay Saunders, Francis Tyers, and Gregor Weber, "Common Voice: A Massively-Multilingual Speech Corpus," in *Proceedings of the Twelfth Language Resources and Evaluation Conference*, Nicoletta Calzolari, Frédéric Béchet, Philippe Blache, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, Eds., Marseille, France, May 2020, pp. 4218–4222, European Language Resources Association.