PHONE-TO-AUDIO ALIGNMENT WITHOUT TEXT: A SEMI-SUPERVISED APPROACH

Jian Zhu[⋆]   *Cong Zhang*[†]   *David Jurgens*[‡]

[⋆]Department of Linguistics, University of Michigan, Ann Arbor, USA
[†]Center for Language Studies, Radboud University, Nijmegen, Netherlands
[‡]School of Information, University of Michigan, Ann Arbor, USA
lingjzhu@umich.edu

ABSTRACT

The task of phone-to-audio alignment has many applications in speech research. Here we introduce two Wav2Vec2-based models for both text-dependent and text-independent phone-to-audio alignment. The proposed Wav2Vec2-FS, a semi-supervised model, directly learns phone-to-audio alignment through contrastive learning and a forward sum loss, and can be coupled with a pretrained phone recognizer to achieve text-independent alignment. The other model, Wav2Vec2-FC, is a frame classification model trained on forced aligned labels that can both perform forced alignment and text-independent segmentation. Evaluation results suggest that both proposed methods, even when transcriptions are not available, generate highly close results to existing forced alignment tools. Our work presents a neural pipeline of fully automated phone-to-audio alignment. Code and pretrained models are available at https://github.com/lingjzhu/charsiu.

Index Terms— forced alignment, phone segmentation, deep learning, automatic speech recognition

1. INTRODUCTION

Aligning phones to audio has been a fundamental task in speech and phonetic research, as many speech analyses depend on knowing the exact timing of phones. There are currently a few existing tools for performing forced alignment over phone sequences [1, 2, 3, 4, 5]. The majority of them are based on the classic HMM system built upon Kaldi [6] or the HTK toolkit [7]. Though neural ASR systems have shown better performance than HMM systems [8], phone-to-audio alignment has not benefited a lot from the predictive power of neural networks. One reason is that model ASR has increasingly shifted towards end-to-end training using loss functions like CTC [9] that disregards precise frame alignment. Only a few works explored using neural networks to perform segmentation of sentences [10] and phones [11, 12, 13]. These works demonstrate great potentials for neural forced alignment, but they still required text transcriptions. Compared to forced alignment, text-independent alignment has received much less attention. While a few

unsupervised or self-supervised phone segmentation methods have been proposed [14, 15, 16, 13], most of them (except [13]) only output phone boundaries but do not jointly predict *both* boundaries and phones, making them less practical in textless alignment. Therefore, we also developed a method to perform alignments directly on audio, as transcriptions might not always be available in many practical applications.

In this study, we present two neural networks for both text-dependent and text-independent phone-to-audio alignment. Our contribution can be summarized as follows. First, we present a semi-supervised model to perform forced alignment, but can also be combined with a phone recognizer for text-independent alignment. It achieves comparable results with four existing forced alignment tools. Secondly, we have also tested a frame classification model for text-independent alignment, which achieved close performance to text-dependent models. Extensive analyses show that the proposed methods still maintain good performance in different settings. In contrast to currently available tools, our work provides speech and language researchers with a strong deep learning based tool to obtain phone segmentations even when textual transcriptions are not available.

2. NEURAL FORCED ALIGNMENT

Given a speech signal $X^S \in \mathbb{R}^{1 \times T_{raw}}$ and a sequence of phonetic symbols $Y^P \in \mathbb{R}^{1 \times N}$, some neural networks are used to encode them into hidden representations Φ, such that $\boldsymbol{Y} = \Phi_S(Y_P)$ and $\boldsymbol{X} = \Phi_P(X_S)$, where $\boldsymbol{X} \in \mathbb{R}^{K \times T} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_t]$ and $\boldsymbol{Y} \in \mathbb{R}^{K \times N} = [\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_n]$, $T \geq N$. We seek to learn a transformation f to encode these features into the monotonic alignment $S = f(\boldsymbol{X}, \boldsymbol{Y})$, so that it aligns each phonetic symbol y_m^P to a subset of consecutive acoustic frames $\boldsymbol{x}_i$: $S = [\boldsymbol{x}_i = y_m^P, \boldsymbol{x}_j = y_m^P, \ldots, \boldsymbol{x}_k = y_n^P]$. If transcriptions are not available, phone labels $\hat{Y}^P$ can be estimated with a phone recognizer from audio.

The proposed neural forced alignment model learns the phone-to-audio alignment through the self-supervised task of reconstructing the quantized embeddings of original speech with both heavily masked speech representations and phone-

mic information [12]. This could be implemented as the same pretraining task of Wav2Vec2 [8]. Meanwhile, we relied on the forward sum loss to guide the attention matrix to be monotonic alignments [17, 18].

2.1. Encoders

We used the pretrained `wav2vec2-base` model [8] with a convolutional layer on top as the speech encoder Φ_S. As the original frame rate of Wav2Vec2 (about 20ms per frame) could be too coarse to derive precise timing, we increased the frame rate by two methods, either by reducing the stride of the last convolutional layer in its feature extractor from 2 to 1 (denoted as `-10ms`), or by upsampling the raw speech signal to 32kHz (denoted as `-32k`). This effectively doubled the frame rate to 98Hz (about 10ms per frame), while allowing the model to load all the pretrained parameters of `wav2vec2-base`.

The phone encoder Φ_T is a reduced version of the BERT model [19] with a convolutional layer on top. It has 4 hidden layers, each of which has 12 self-attention heads of 32 hidden dimensions. The model was pretrained with masked language modeling [19] on phone sequences generated from the whole Book Corpus [20], which amounts to more than 2 billion of phone tokens.

2.2. Loss function

The model learns a matrix $\boldsymbol{A} \in \mathbb{R}^{N \times T}$ that aligns the phone sequence $\boldsymbol{Y}$ and the masked speech representation $\hat{\boldsymbol{X}}$.

$$\boldsymbol{D}_{ij} = sim(\boldsymbol{y}_i, \hat{\boldsymbol{x}}_j) \tag{1}$$

$$\boldsymbol{A} = softmax(\boldsymbol{D}, \texttt{dim=0}) \tag{2}$$

$$\boldsymbol{H} = concatenate[\hat{\boldsymbol{X}}, \boldsymbol{A}^T \boldsymbol{Y}] \tag{3}$$

For the hidden states $\boldsymbol{H} \in \mathbb{R}^{2K \times T} = [\boldsymbol{h}_1, \ldots, \boldsymbol{h}_T]$ at each time step, the model maximizes the similarity between each $\boldsymbol{h}_t$ and the quantized version of the acoustic feature at the same time step. The quantized embeddings can be selected from a codebook using Gumbel softmax parameterized by a neural network, which is detailed in the Wav2Vec2 paper [8]. Let $\boldsymbol{Q}_t$ represents the set consisting of a true quantized embedding and $n = 50$ negative samples at step t. The loss function can be formulated as follows.

$$\mathcal{L}_m = \frac{1}{T} \sum_{t=1}^{T} - \log \frac{\exp\left(sim(\boldsymbol{h}_t, \boldsymbol{q}_t)/\kappa\right)}{\sum_{\boldsymbol{q}_k \sim \boldsymbol{Q}_t} \exp(sim(\boldsymbol{h}_t, \boldsymbol{q}_k)/\kappa)} \tag{4}$$

The masking is performed with spectral augmentation with both time and feature masking [21]. If the time masking probability is too low, the model might simply ignore the phone information. However, if the time masking probability is too high, it will create a gap between training and inference inputs, as the input audio for inference is usually not

corrupted. We sampled the time-masking probability $p/100$ for each batch from a discrete uniform distribution of $[p_l, p_h]$. The original weights of the Wav2Vec2 quantizer were kept fixed throughout training.

Yet the attention matrices might not necessarily align acoustic frames and phone tokens. Therefore we added the forward-sum loss used in HMM systems to constrain the attention matrix to be monotonic and diagonal [17, 18].

$$\mathcal{L}_{FS} = - \sum_{\boldsymbol{X}, \boldsymbol{Y} \in S} \log P(\boldsymbol{Y}|\boldsymbol{X}) \tag{5}$$

where S is the optimal alignment. All probability quantities are estimated in the attention matrix $\boldsymbol{A}$. This loss function can be implemented using the off-the-shelf PyTorch CTC loss by setting the probability of the blank symbol to an impossibly low value. We adapted the implementation from [17, 18]. So the final loss is a weighted sum of the two loss functions $\mathcal{L} = \mathcal{L}_m + \lambda \mathcal{L}_{FS}$, where λ was set to 1.

The alignment is notoriously hard to train, since early deviations from the ground truth only get reinforced in subsequent iterations [22]. Techniques such as guided attention loss [23] or diagonal prior [17, 18] or external supervision with available alignments [24] have been used to enable the formation of text-to-speech alignment. To facilitate alignment learning, we train the model incrementally in the curriculum learning paradigm [25] on increasingly longer audio samples. Like noisy student training [26] and the iterative training in Kaldi [6], a series of models are generated in the process.

1. Initialize model M with pretrained weights M_0. Set $M = M_0$.
2. Partition the length-sorted data C into k different chunks $C_1, C_2, \ldots, C_k$ by duration, such that each chunk contains audio longer than the previous chunk. Then train M on each chunk sequentially with a frame shift of 20ms.
3. Upsample the frame shift of M to 10ms. Retrain M using the same data and procedure as in 2.
4. (*Optional*) Train a frame classification model M' from scratch with alignments generated from the last step. Set $M = M'$ and go back to 2.

Reinitialization with a frame classification model speeds up the convergence as it has been noted that a prior is essential for obtaining good alignment [22]. As the learned attention alignments are sharp, the decoding can be performed simply using the `argmax` function without resorting to the Viterbi decoding. This forced alignment is referred to as `W2V2-FS`.

3. FRAME CLASSIFICATION MODEL

For text-independent alignment, we fine-tuned a Wav2Vec2 model to perform frame-wise phone classification (referred to as `W2V2-FC-Libris`). The training labels were alignments of the 960h *Librispeech* [27] obtained through MFA used in [28]. The alignments were discretized into 10ms

Table 1. Evaluation results of text-dependent alignment

Model	P	R	F1	R-val	Overlap
FAVE	0.57	0.59	0.58	0.64	74.3%
MFA-Libris	0.61	0.61	0.61	0.67	73.5%
MFA	0.62	0.63	0.63	0.68	75.0%
Gentle	0.49	0.46	0.48	0.56	67.7%
WebMAUS	**0.70**	**0.70**	**0.70**	**0.75**	78.8%
W2V2-FC-20ms-Libris	0.49	0.47	0.48	0.56	73.8%
W2V2-FC-10ms-Libris	0.57	0.54	0.55	0.62	76.4%
W2V2-FC-32k-Libris	0.66	0.63	0.64	0.69	79.3%
W2V2-FS-20ms	0.47	0.49	0.48	0.55	71.6%
W2V2-FS-10ms	0.68	0.68	0.68	0.73	**80.4%**
W2V2-FS-32k	0.63	0.65	0.64	0.69	79.3%
Pretrained G2P converter					
W2V2-FS-20ms	0.40	0.42	0.41	0.49	65.1%
W2V2-FS-10ms	0.56	**0.58**	0.57	0.63	72.5
W2V2-FC-32k-Libris	**0.58**	0.57	0.58	**0.64**	**73.0%**
Phone set adaptation (TIMIT-61)					
W2V2-FS-20ms	0.49	0.53	0.51	0.57	70.5%
W2V2-FS-10ms	**0.66**	**0.70**	**0.68**	**0.72**	**79.7%**

Table 2. Evaluation results of text-independent alignment

Model	P	R	F1	R-val	Overlap
W2V2-CTC-10ms	0.31	0.29	0.30	0.42	43.9%
W2V2-CTC-20ms	0.31	0.30	0.31	0.42	46.6%
Phone recognition + W2V2-FS					
W2V2-FS-20ms	0.40	0.42	0.41	0.48	64.2%
W2V2-FS-10ms	0.56	0.58	0.57	0.63	71.5%
W2V2-FC-32k-Libris	0.57	0.57	0.57	0.64	72.2%
Direct inference					
W2V2-FC-20ms-Libris	0.57	0.59	0.58	0.63	72.7%
W2V2-FC-10ms-Libris	0.55	0.58	0.56	0.62	72.5%
W2V2-FC-32k-Libris	**0.60**	**0.63**	**0.61**	**0.66**	**74.3%**

or 20ms frames and the model was trained to classify each frame into a phone category using the cross-entropy loss. During inference, the phone alignment was acquired through model predictions directly with audio, but forced alignment was also performed using a Dynamic Time Warping (DTW) algorithm based on an output probability matrix and its transcription. We also trained additional frame classification models (W2V2-FC) on the alignments generated from the semi-supervised method.

4. EXPERIMENTS

For the English dataset, we used the Librispeech [27] the 1600h *Common Voice* 6.1 [29]. The TIMIT dataset with human annotations [30] was used for evaluation. The original TIMIT 61 phone annotations were collapsed into the 39 CMU phone set. The flap [ɾ] DX did not have a one-to-one mapping, so we kept all occurrences. While this slightly degraded the overall accuracy, this was the same for all forced alignment methods under comparison except WebMAUS, which used more phones than those in the CMU phone set. All data and their train/test partitions were based on the Hugging-Face datasets package [31]. All textual transcriptions were converted into phone sequences using an open source grapheme-to-phoneme (G2P) converter [32].

In our experiment, we ran four iterations of training. We set the time masking probability to $[0.05, 0.2]$ for the 20ms models and $[0.05, 0.4]$ for 10ms models. The first two iterations were trained on Common Voice. We partitioned the Common Voice dataset into three sets $\{C1, C2, C3\}$, $C1$ containing sentences below 3 seconds, $C2$ with sentences between 3-5 seconds and $C3$ consisting of sentences up to 10 seconds. The third iteration was trained on Librispeech and in the fourth iteration, the model was fine-tuned on the TIMIT train set. The model generally converged within a thousand training steps so training on full dataset was not needed. Annotated boundary information was never used during training. Evaluation was performed on the TIMIT test set, which does

not overlap with any of the training data. The proposed model was coded in pytorch based on the Wav2Vec2 implementations in the transformers package. By default, we used an effective batch size of 32 and the Adam optimizer with a learning rate of 1e-6 and rate decay of 1e-6. The training was done distributedly on three GPUs with 12 GB of memory.

We compared our method with four publicly available forced aligners: Montreal Forced Aligner (MFA) [1], Penn Forced Aligner (FAVE) [3], WebMAUS [4] and Gentle [5]. Specifically for MFA, we tested its performance both using a pretrained Librispeech model and performing forced alignment from scratch on TIMIT.

5. EVALUATION

We evaluated the forced alignment results with precision, recall, F1, and R-value [14]. For each predicted phone boundary, if the timing was within tolerance τ and the predicted phone matches, it was considered a hit. As each boundary marked the onset and the offset of consecutive phones, we only evaluated the phone onsets with a tolerance of 20ms. In addition, we also measured the percentage of correctly predicted frame labels at a 10ms time scale.

In Table 1, all methods can perform forced alignment with decent accuracy, though Gentle seems to lag slightly behind other methods. The proposed forced alignment method also shows comparable performance, comparable to all existing forced alignment methods, if not better. In general, increasing the frameshift from 20ms to 10ms improved the alignment accuracy. However, phones in continuous speech often undergo a variety of phonetic processes such as deletion and insertion, resulting in pronunciation variants deviating from dictionary pronunciations. Yet most G2P converters can only provide dictionary pronunciations, which may lead to decreased performance. To shed light on this issue, we tested the influence of a pretrained G2P converter. In Table 1, it is clear that using the dictionary pronunciations decreases the alignment performance by a significant margin, yet the performance is still close to current forced aligners.

As different speech varieties may have varying phone inventories (like American English vs. Received Pronunciation), it would be desirable for the pretrained W2V2-FS model to adapt to different phone inventories. To test this,
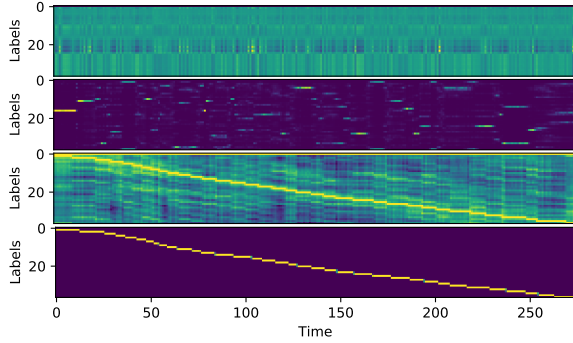
Fig. 1. Sample alignments. From top to bottom: 1) No curriculum training; 2) No $\mathcal{L}_{FS}$; 3) No $\mathcal{L}_m$; 4) Full model.

Table 3. Evaluation of `W2V2` models from different iterations

Model	Iteration	Training Data	R-val	Overlap
FS-10ms	1	Common Voice	0.41	60.7%
FC-10ms(w/o text)	1	Common Voice	0.37	55.8%
FS-10ms	2	Common Voice	0.51	68.1%
FC-10ms(w/o text)	2	Common Voice	0.53	66.2%
FS-10ms	3	Librispeech	0.60	74.1%
FC-10ms(w/o text)	3	Librispeech	0.52	68.1%
FC-10ms(w/ text)	3	Librispeech	**0.63**	**75.3%**

we can simply fine-tune the model after modifying the phone embeddings of the text encoder. For TIMIT, we expanded the existing 39 phone embeddings of the original `W2V2-FS` model to include the full TIMIT 61 phone set while keeping the rest of the pretrained weights intact, and fine-tuned the `W2V2-FS` model. The model converged quickly after a few hundred iterations and achieved comparable performance as the original model (see last two rows in Table 1). Yet `W2V2-FC` models might be less flexible in this regard, as it requires pre-existing alignments to train.

For text-independent alignment, we trained a phone recognizer on both Librispeech and Common Voice by fine-tuning `wav2vec2-base` using the CTC loss. This phone recognizer was used to derive phone transcriptions first before performing forced alignment with `W2V2-FS`. Using a phone recognizer works reasonably well (Table 2), as the alignment is only 1% worse than using a pretrained G2P converter. As expected, the alignments generated by the CTC loss alone deviated significantly from the ground truth, since the CTC loss does not encourage time-synchronous alignment. In contrast, `W2V2-FC` models achieved better performance (Table 2) than other text independent models, highly comparable to that of text-dependent alignment methods. This proves that the frame classification model used in HMM systems are still robust with neural networks. Overall, these encouraging results suggest that both text independent models are practical means to derive phone-to-audio alignments.

When the errors of G2P are taken into account, our forced alignment methods might not possess a strong advantage over existing tools. The discrepancy could be accounted for by two causes. First, silence poses challenges to forced alignment as

it is not encoded in the transcription. Without preprocessing silence, phones are often misaligned when a long silent interval is present. Secondly, we did not explicitly model pronunciation variants. However, HMM-based systems [6] often have built-in methods to alleviate these two issues. This will be the next step for our research.

6. ABLATION ANALYSIS

We performed ablation analysis to examine the effectiveness of the `W2V2-FS` model. As shown in Fig 1, both the contrastive loss and the forward-sum loss are necessary to facilitate the learning of sharp attention alignment. If the audio and the transcription were aligned using only the forward sum loss, the learned alignment was noisy. If either the curriculum learning or the contrastive loss is removed, the model generally fails to converge to a diagonal alignment pattern.

Multiple iterations of training are generally needed to boost performance (see Table 3), as initial alignments tend to be suboptimal, especially for noisy datasets like Common Voice. During training, we found that the forced alignment model was *very* sensitive to initialization, and poor alignment was reinforced but not corrected in subsequent training. So using a frame classification to provide a good prior estimation was important. Audio quality had a large impact on the quality of alignment. Training on TIMIT or Librispeech-clean from scratch ended up with better alignments, faster convergence rates, and fewer iterations than training on the crowd-sourced Common Voice. Table 3 illustrates the variations of model performance when noisy data and audio domain mismatch are both present, though the performance gaps can be narrowed through fine-tuning.

7. DISCUSSIONS AND CONCLUSIONS

In this study, we present two deep learning-based methods to align audio to phones and achieve comparable performance against several existing forced alignment tools. These models can be combined as a pipeline to bootstrap phone labels from naturalistic audio data in the wild, which are massive but remain under-exploited. This will have implications for speech corpus creation, phonetic research, and interdisciplinary studies. Given the good performance of `W2V2-FC-Libris`, our method can also be combined with MFA[1] to train a large-scale text-independent frame classification model to get rid of text in inference. The W2V2 models can be accelerated by harnessing the computing power of GPU. For example, in the inference mode, our model aligned the 1600 hours Common Voice dataset in about 10 hours on a 2080Ti GPU without batching. While only tested on phone segmentation, the model also performed well on related tasks such as word or sentence segmentation. In the future, we will continue to improve the alignment accuracy and extend it to different languages.

8. REFERENCES

[1] M. McAuliffe et al., "Montreal forced aligner: Trainable text-speech alignment using kaldi.," in *INTERSPEECH*, 2017, vol. 2017, pp. 498–502.

[2] K. Gorman et al., "Prosodylab-aligner: A tool for forced alignment of laboratory speech," *Canadian Acoustics*, 2011.

[3] I. Rosenfelder et al., "FAVE (forced alignment and vowel extraction) program suite," *URL http://fave. ling. upenn. edu*, 2011.

[4] T. Kisler et al., "Signal processing via web services: the use case webmaus," in *Digital Humanities Conference 2012*, 2012.

[5] R. Ochshorn and M. Hawkins, "Gentle," https://lowerquality.com/gentle/, 2017.

[6] D. Povey et al., "The kaldi speech recognition toolkit," in *ASRU*, 2011.

[7] S. Young et al., "The HTK book," *Cambridge university engineering department*, vol. 3, no. 175, pp. 12, 2002.

[8] A. Baevski et al., "wav2vec 2.0: A framework for self-supervised learning of speech representations," in *NeurIPS*, 2020.

[9] A. Graves et al., "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *ICML*, 2006.

[10] L. Kürzinger et al., "CTC-segmentation of large corpora for german end-to-end speech recognition," in *International Conference on Speech and Computer*, 2020.

[11] M. Kelley and B. Tucker, "A comparison of input types to a deep neural network-based forced aligner," in *INTERSPEECH*, 2018.

[12] K. Schulze-Forster et al., "Joint phoneme alignment and text-informed speech separation on highly corrupted speech," in *ICASSP*, 2020.

[13] Y. Teytaut and A. Roebel, "Phoneme-to-Audio Alignment with Recurrent Neural Networks for Speaking and Singing Voice," in *INTERSPEECH*, 2021.

[14] F. Kreuk et al., "Self-Supervised Contrastive Learning for Unsupervised Phoneme Segmentation," in *INTERSPEECH*, 2020.

[15] H. Kamper and B. van Niekerk, "Towards unsupervised phone and word segmentation using self-supervised vector-quantized neural networks," *arXiv preprint arXiv:2012.07551*, 2020.

[16] S. Bhati et al., "Segmental contrastive predictive coding for unsupervised word segmentation," *arXiv preprint arXiv:2106.02170*, 2021.

[17] K. Shih et al., "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.

[18] R. Badlani et al., "One tts alignment to rule them all," *arXiv preprint arXiv:2108.10447*, 2021.

[19] J. Devlin et al, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *NAACL*, 2019, pp. 4171–4186.

[20] Y. Zhu et al., "Aligning books and movies: Towards story-like visual explanations by watching movies and reading books," in *ICCV*, 2015.

[21] D. Park et al., "SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition," in *INTERSPEECH*, 2019.

[22] A. Zeyer et al., "CTC in the Context of Generalized Full-Sum HMM Training," in *INTERSPEECH*, 2017.

[23] H. Tachibana, "Efficiently trainable text-to-speech system based on deep convolutional networks with guided attention," in *ICASSP*, 2018.

[24] Y. Ren et al., "Fastspeech 2: Fast and high-quality end-to-end text to speech," in *ICLR*, 2020.

[25] Y. Bengio et al., "Curriculum learning," in *ICML*, 2009.

[26] D Park et al., "Improved noisy student training for automatic speech recognition," in *INTERSPEECH*, 2020.

[27] V. Panayotov et al., "Librispeech: an asr corpus based on public domain audio books," in *ICASSP*, 2015.

[28] L. Lugosch et al., "Speech Model Pre-Training for End-to-End Spoken Language Understanding," in *ITERSPEECH*, 2019.

[29] R. Ardila et al., "Common voice: A massively-multilingual speech corpus," in *LREC*, 2020.

[30] J. Garofolo et al., "DARPA TIMIT acoustic-phonetic continous speech corpus CD-ROM. NIST speech disc 1-1.1," *NASA STI/Recon technical report*, 1993.

[31] Q Lhoest et al., "Datasets: A community library for natural language processing," *arXiv preprint arXiv:2109.02846*, 2021.

[32] Kyubyong Park and Jongseok Kim, "g2pE," https://github.com/Kyubyong/g2p, 2019.