Learning to Speak from Text: Zero-Shot Multilingual Text-to-Speech with Unsupervised Text Pretraining

Takaaki Saeki[1] , **Soumi Maiti**[2] , **Xinjian Li**[2] , **Shinji Watanabe**[2] ,
Shinnosuke Takamichi[1] and **Hiroshi Saruwatari**[1]
[1]The University of Tokyo, Japan
[2]Carnegie Mellon University, USA
takaaki_saeki@ipc.i.u-tokyo.ac.jp, {smaiti, swatanab}@andrew.cmu.edu

Abstract

While neural text-to-speech (TTS) has achieved human-like natural synthetic speech, multilingual TTS systems are limited to resource-rich languages due to the need for paired text and studio-quality audio data. This paper proposes a method for zero-shot multilingual TTS using text-only data for the target language. The use of text-only data allows the development of TTS systems for low-resource languages for which only textual resources are available, making TTS accessible to thousands of languages. Inspired by the strong cross-lingual transferability of multilingual language models, our framework first performs masked language model pretraining with multilingual text-only data. Then we train this model with a paired data in a supervised manner, while freezing a language-aware embedding layer. This allows inference even for languages not included in the paired data but present in the text-only data. Evaluation results demonstrate highly intelligible zero-shot TTS with a character error rate of less than 12% for an unseen language.

1 Introduction

Recent advances in neural text-to-speech synthesis (TTS) [Li *et al.*, 2019b; Kim *et al.*, 2021] have yielded significant improvements in naturalness and speech quality. However, the data-intensive nature and the requirement of paired text and studio-quality audio data have limited multilingual TTS systems to resource-rich languages, which are small portions of the more than 6,000 languages in the world [Gordon Jr, 2005]. To address the limitation, current research in multilingual TTS aims not only to exploit resource-rich languages [Zen *et al.*, 2012; Li and Zen, 2016] but also to build models for low-resource languages [Prakash *et al.*, 2019].

Previous work has addressed low-resource TTS by using untranscribed speech data with vector-quantized variational autoencoder (VQ-VAE) [Zhang and Lin, 2020] or automatic speech recognition (ASR) models [Ni *et al.*, 2022]. Another study [Saeki *et al.*, 2022b] has built a massively multilingual TTS model jointly using paired TTS, paired ASR, unpaired speech, and unpaired text data. However, these approaches still rely on speech data for the target languages and face the challenge of data collection, when audio recordings for these languages are hard to obtain. In this study, we focus on the use of a text-only data for multilingual TTS as shown in Fig. 1. Previous research [Wu and Dredze, 2019; Pires *et al.*, 2019] has shown the strong cross-lingual transferability of multilingual language models such as multilingual BERT [Devlin *et al.*, 2019] in natural language processing (NLP) tasks. By leveraging multilingual pretraining, the model can generalize to other languages, even if it has never seen the target data in those languages. Our work applies the framework of multilingual masked language model (MLM) pretraining to TTS, with the goal of achieving zero-shot cross-lingual transfer of pronunciation and prosody. Zero-shot TTS using text data enables the development of TTS systems for languages where only textual resources are available, which potentially opens up TTS to thousands of languages [Ebrahimi and Kann, 2021; Li *et al.*, 2022].

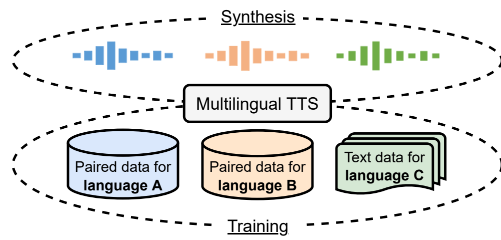

Figure 1: Our concept. We aim to build TTS model on languages for which only text data is available, to support low-resource languages.

In this paper, we propose a multilingual TTS framework that leverages unsupervised text pretraining. Fig. 2 illustrates the proposed framework. We use a typical end-to-end TTS architecture consisting of token embedding, encoder, and decoder. Our model also has a language-aware embedding layer, which includes the token embedding layer, a language embedding layer, and a bottleneck layer. As shown in Fig. 2(a), we first pretrain the language-aware embedding layer and the encoder of the TTS model with multilingual text data. We then fine-tune the encoder and decoder of the TTS model with paired data, while the language-aware embedding layer is frozen, as illustrated in Fig. 2(b). This allows zero-shot TTS for a language not included in the paired data but present in the text data, as shown on the right in Fig. 2(c).

Our contributions are as follows. 1) We propose a zero-shot

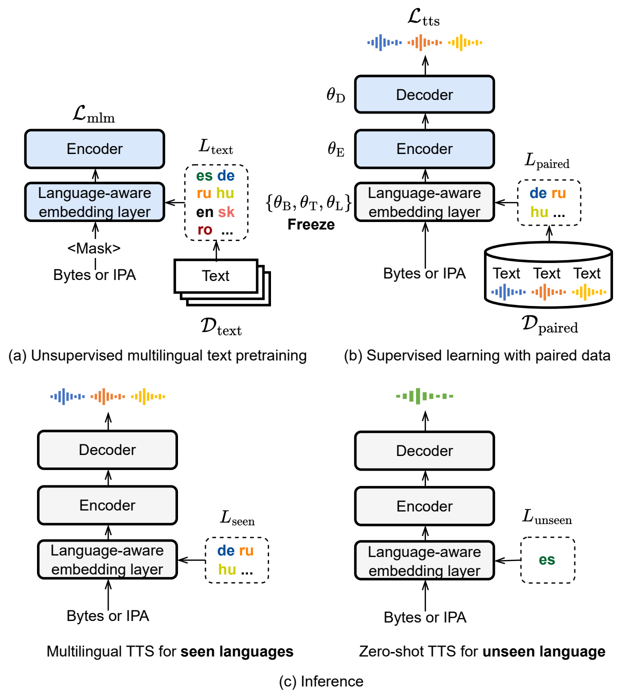

Figure 2: Proposed framework. (a) We perform MLM pretraining on multilingual text data and then (b) train TTS model on paired data with frozen language-aware embedding layer. (c) Zero-shot TTS is performed with language IDs that are not included in paired data.

multilingual TTS framework that achieves highly intelligible TTS for an unseen language, resulting in a character error rate of less than 12%. 2) Our method also improves TTS for seen languages, resulting in byte-based models without grapheme-to-phoneme (G2P) modules that outperform the phoneme-based baselines. 3) Our ablation studies provide additional insights, including the effectiveness of the frozen language-aware embedding layer. The experiments were conducted on public datasets and the implementation is available[1]. We encourage readers to listen to our audio samples[2].

2 Method

Our model has a typical neural TTS model architecture consisting of token embedding, encoder, and decoder. First, we use MLM pretraining with multilingual text data to learn cross-lingual representations. Then we perform supervised learning with paired data to learn the mapping from linguistic features to speech features. The model performs inference even for languages that are not present in the paired data.

2.1 Unsupervised Multilingual Text Pretraining

Fig. 2(a) illustrates the unsupervised pretraining method. It uses multilingual text data consisting of languages that are not included in the paired data. Let $X = (x_n \in V | n = 1, \cdots, N)$ denote the input text token sequence of length N, where V denotes a vocabulary constructed for pretraining. We define $\mathcal{D}_{\text{text}}$ as the text dataset. Let L_{text} denote the set of language IDs included in $\mathcal{D}_{\text{text}}$. First, the masked token sequence X^{m} and a language ID $l_{\text{text}} \in L_{\text{text}}$ are fed to the model. Let the token embedding sequence and language embedding be $Z^{\text{m}} = (\boldsymbol{z}_n^{\text{m}} \in \mathbb{R}^d | n = 1, \cdots, N)$ and $\boldsymbol{e}_l \in \mathbb{R}^d$, respectively. The embedding layers output Z^{m} and $\boldsymbol{e}_l$ as:

$$Z^{\text{m}} = \text{Embed}(X^{\text{m}}; \theta_{\text{T}}), \qquad \boldsymbol{e}_l = \text{Embed}(l_{\text{text}}; \theta_{\text{L}}), \quad (1)$$

where θ_{T} and θ_{L} denote the model parameters of the token embedding and language embedding layers, respectively. Then the token and language embeddings obtained in Eq. (1) are added and fed to a bottleneck layer to project them into a hidden input vector. Let $H_{\text{in}} = (\boldsymbol{h}_{\text{in},n} \in \mathbb{R}^d | n = 1, \cdots, N)$ and $H_{\text{out}} = (\boldsymbol{h}_{\text{out},n} \in \mathbb{R}^d | n = 1, \cdots, N)$ denote hidden vectors in the encoder input and output, respectively. Then the conditional probability $p(X|X_{-\Pi})$ is computed as:

$$H_{\text{in}} = \text{Bottleneck}(Z^{\text{m}} + \boldsymbol{e}^l; \theta_{\text{B}}), \quad (2)$$

$$H_{\text{out}} = \text{Encoder}(H_{\text{in}}; \theta_{\text{E}}), \quad (3)$$

$$p(X|X_{-\Pi}) = \text{Softmax}(\text{PredictionNet}(H_{\text{out}}; \theta_{\text{P}})), \quad (4)$$

where θ_{B}, θ_{E}, θ_{P} denote the model parameters of the bottleneck layer, the encoder and a prediction network, respectively. In Eq. (4), $\text{Softmax}(\cdot)$ denotes a softmax function. We define the network with the model parameters $\{\theta_{\text{B}}, \theta_{\text{T}}, \theta_{\text{L}}\}$ as **language-aware embedding layer**, which jointly embeds the token sequence X and the language ID l_{text} as in Eq. (1) and (2). Let $\Pi = (\pi_k \in \mathbb{N} | k = 1, \cdots, K)$ be the indexes of the masked tokens of length K. With the probability computed in Eq. (4), the training objective can be defined as:

$$\mathcal{L}_{\text{mlm}} = \frac{1}{K} \sum_{k=1}^{K} \log p(x_{\pi_k} | X^{\text{m}}),$$
$$\{\hat{\theta}_{\text{E}}, \hat{\theta}_{\text{B}}, \hat{\theta}_{\text{T}}, \hat{\theta}_{\text{L}}\} = \underset{\theta_{\text{E}}, \theta_{\text{B}}, \theta_{\text{T}}, \theta_{\text{L}}}{\arg\min} \ \mathcal{L}_{\text{mlm}}. \quad (5)$$

We use UTF-8 bytes or International Phonetic Alphabet (IPA) symbols for the input token sequence X. For each token type, the vocabulary V is constructed from $\mathcal{D}_{\text{text}}$, which includes a start/end of sentence token ([`SOS/EOS`]). We extracted International IPA sequences using an open-source toolkit[3]. To obtain the masked token X^{m}, we use the same masking ratio and category as in the original BERT pretraining [Devlin *et al.*, 2019] for each token type. Randomly, 12 % of the tokens are replaced with the [`MASK`] token, and 1.5 % of them are replaced with random tokens. Also, 1.5 % of the tokens are left unchanged and $\mathcal{L}_{\text{mlm}}$ is computed as in Eq. (5) for those 15 % of tokens that have indices Π.

2.2 Supervised Learning with Paired Data

Fig. 2(b) illustrates the supervised learning of the TTS model with paired data. We define the paired data and the set of language IDs as $\mathcal{D}_{\text{paired}}$ and L_{paired}, respectively. Note that we assume $L_{\text{paired}} \subset L_{\text{text}}$. Let $Y = (\boldsymbol{y}_t \in \mathbb{R}^D | t = 1, \cdots, T)$ denote the speech feature sequence with the length of T. We first initialize the model parameters $\{\theta_{\text{E}}, \theta_{\text{B}}, \theta_{\text{T}}, \theta_{\text{L}}\}$ with

[1]https://github.com/Takaaki-Saeki/zm-text-tts

[2]https://takaaki-saeki.github.io/zm-tts-text_demo

[3]https://github.com/espeak-ng/espeak-ng

those obtained in the pretraining described in § 2.1. Let θ_{D} denote the model parameter of the decoder. The speech features are predicted with teacher forcing as:

$$H_{\mathrm{out}} = \mathrm{Encoder}(\mathrm{Bottleneck}(Z + \boldsymbol{e}^{l})), \tag{6}$$

$$\hat{Y} = \mathrm{Decoder}(H_{\mathrm{out}}, Y; \theta_{\mathrm{D}}), \tag{7}$$

where Z is the unmasked token embedding sequence. Note that the unmasked token sequence is used in Eq. (6), while the masked token sequence is used in Eq. (2) Let $\mathcal{L}_{\mathrm{tts}}(\hat{Y}, Y)$ denote the training objective of the TTS model. Then we consider two types of schemes.

Updating language-aware embedding layer. We only freeze the parameter of the language embedding layer θ_{L} while updating the rest of the parameters. Therefore the trainable model parameters can be written as

$$\{\hat{\theta}_{\mathrm{D}}, \hat{\theta}_{\mathrm{E}}, \hat{\theta}_{\mathrm{B}}, \hat{\theta}_{\mathrm{T}}\} = \underset{\theta_{\mathrm{D}}, \theta_{\mathrm{E}}, \theta_{\mathrm{B}}, \theta_{\mathrm{T}}}{\arg\min} \; \mathcal{L}_{\mathrm{tts}}(\hat{Y}, Y). \tag{8}$$

Previous work has confirmed that multilingual BERT has high cross-lingual transferability for various NLP tasks [Wu and Dredze, 2019]. This scheme corresponds to a simple finetuning of BERT [Wu and Dredze, 2019], which updates all the parameters during training for the downstream tasks[4].

Freezing language-aware embedding layer. We freeze the bottleneck layer and the token embedding layer along with the language embedding, updating the encoder and decoder. The training process can be written as

$$\{\hat{\theta}_{\mathrm{D}}, \hat{\theta}_{\mathrm{E}}\} = \underset{\theta_{\mathrm{D}}, \theta_{\mathrm{E}}}{\arg\min} \, \mathcal{L}_{\mathrm{tts}}(\hat{Y}, Y). \tag{9}$$

In contrast to the scheme represented in Eq. (8), the scheme in Eq. (9) preserves the parameters of the language-aware embedding layer to facilitate cross-lingual transfer. In the evaluation, we use the scheme formulated in Eq. (9), except for the ablation study in § 3.4.

2.3 Inference

Let L_{syn} denote the set of language IDs used for inference. The text token sequence X and the language ID $l_{\mathrm{syn}} \in L_{\mathrm{syn}}$ are fed to the model as in Eq. (1), and the encoder output is predicted as in Eq. (6). Unlike Eq. (7), the speech features are predicted as:

$$\hat{Y} = \mathrm{Decoder}(H_{\mathrm{out}}; \theta_{\mathrm{D}}). \tag{10}$$

The output waveform is obtained by feeding the predicted features $\hat{Y}$ to a pretrained neural vocoder.

Fig. 2(c) illustrates the inference process. The left and right sides of the figure show the typical multilingual TTS and our zero-shot TTS. Previous work [Li *et al.*, 2019a] has typically assumed *seen* languages, and the inference is performed with the language IDs $L_{\mathrm{seen}} \subset L_{\mathrm{paired}}$. However, it is challenging to perform TTS for *unseen* languages $L_{\mathrm{unseen}} \cap L_{\mathrm{paired}} = \emptyset$. While other work [Saeki *et al.*, 2022b] has built a massively multilingual TTS model that even achieves zero-shot TTS from ASR data, it uses paired data for the target languages. Our work attempts to only use the linguistic knowledge to improve the zero-shot TTS. Thus, the inference process is written as $L'_{\mathrm{unseen}} \cap L_{\mathrm{paired}} = \emptyset$ and $L'_{\mathrm{unseen}} \subset L_{\mathrm{text}}$. In the evaluation, we denote the inference with L_{unseen} and L'_{unseen} as *Fully zero-shot TTS* and *Text-seen zero-shot TTS*, respectively. *Fully zero-shot TTS* performs zero-shot TTS without pretraining as in the IPA-based previous method [Staib *et al.*, 2020], which is the baseline method in our evaluations.

[4] We freeze the language embedding layer to address the mismatch between language embedding of seen and unseen languages.

2.4 Model Architecture

Our model is an autoregressive TTS model based on Transformer TTS [Li *et al.*, 2019b], which has also been used in the previous work on byte-based multilingual TTS [He *et al.*, 2021]. During the supervised learning described in § 2.2 and inference described in § 2, we use x-vector [Snyder *et al.*, 2018] for the speaker embedding and add it to the encoder output through a projection layer. During supervised learning, we use the average x-vectors computed from the training data. For evaluation purposes, we perform zero-shot synthesis with the average x-vector from the test data of the target language and feed it to the model. Note that we also conduct the evaluation with x-vectors from seen languages.

For the bottleneck layer with θ_{B}, we use a residual network consisting of Layer Normalization [Ba *et al.*, 2016], down projection, ReLU [Nair and Hinton, 2010], and up projection with the residual connection, which is used in previous work on language adaptation [Bapna *et al.*, 2019].

3 Experimental Evaluations

3.1 Experimental Setting

Dataset

We carried out all the evaluations with publicly available datasets. Table 1 shows the sizes of the data for each language. For the unsupervised text pretraining described in § 2.1, we used transcripts from VoxPopuli [Wang *et al.*, 2021], M-AILABS [Munich Artificial Intelligence Laboratories GmbH, 2017], and CSS10 [Park and Mulc, 2019], resulting in a total of about 2.8 GB of spoken text across 19 languages. We used CSS10 for the supervised learning described in § 2.2, and we selected seven European languages as the seen languages, with Spanish as the unseen language. The paired data consisted of one speaker per language. It should be noted that Spanish is not actually a low-resource language, but we chose to use it for evaluation purposes in order to 1) compare our zero-shot TTS methods with the oracle methods using the paired data for the target language and 2) ensure a sufficient number of evaluators for the subjective evaluation. We used 5 and 100 utterances as dev and test sets, respectively, with the remaining data used for training.

Training Details

The sampling rate was set to 16 kHz. An 80-dimension of mel filter bank, 1024 samples of FFT length, and 256 samples of frame shit were used for speech analysis. For the pretraining described in § 2.1, we trained the model for 1.2M iterations using the Noam optimizer [Vaswani *et al.*, 2017] with the learning rate and warm-up step set to 1.0 and 10000, respectively. For the TTS model described in

Languages	Code	Text-only data	Paired data	
			Text	Audio
Seen languages for evaluation L_{seen}				
German	de	359MB	0.73MB	16.13h
French	fr	372MB	0.94MB	19.15h
Dutch	nl	336MB	0.75MB	14.10h
Finnish	fi	308MB	0.47MB	21.36h
Hungarian	hu	104MB	0.51MB	10.53h
Russian	ru	4.9MB	1.5MB	10.00h
Greek	el	0.39MB	0.39MB	4.13h
Unseen language for evaluation L_{unseen}				
Spanish	es	345MB	0.0MB (1.2MB)	0.00h (23.81h)
Languages not included in CSS10				
English	en	338MB		
Estonian	et	87MB		
Croatian	hr	2.0MB		
Italian	it	334MB		
Lithuanian	lt	89MB		
Polish	pl	102MB		
Romanian	ro	67MB		
Slovak	sk	94MB		
Slovenian	sl	81MB		

Table 1: Amount of text-only and paired data for each language. Parentheses indicate amount of original data in CSS10.

§ 2.4, we used a 6-block Transformer encoder [Vaswani *et al.*, 2017] and a 6-block Transformer decoder, with a postnet consisting of five convolutional layers with a kernel size of five. The attention dimension and the number of attention heads were set to 512 and 8, respectively. For the bottleneck layer described in § 2.4, we set the hidden dimension after the down projection to 256. The PredictionNet in Eq. (4) consisted of a linear layer, a GELU activation function [Hendrycks and Gimpel, 2016], Layer Normalization, and a linear layer with the hidden dimension of 512. We also used guided attention loss [Tachibana *et al.*, 2018] to improve the training efficiency. For the supervised learning described in § 2.2, we trained the models for 2.47M iterations (200 epochs). The Noam optimizer was used with the warm-up step of 50000. For the neural vocoder, we trained HiFi-GAN [Kong *et al.*, 2020] for 2M iterations with LibriTTS [Zen *et al.*, 2019], VCTK [Veaux *et al.*, 2017], and CSS10. For the x-vector described in § 2.4, we used a model trained on VoxCeleb1 and VoxCeleb2 [Nagrani *et al.*, 2017] published in SpeechBrain [Ravanelli *et al.*, 2021]. We used ESPnet2-TTS [Watanabe *et al.*, 2018; Hayashi *et al.*, 2021] for the implementation.

Baselines

We developed baseline models without the pretraining.

Seen language. *Monolingual:* We trained a model for each language independently. Our preliminary study found that Transformer TTS was unstable[5] and could not synthesize intelligible speech in the monolingual condition due to the lack of training data. Therefore, we used Tacotron2 [Shen *et al.*, 2018] only for the monolingual models, as in the original paper of the dataset [Park and Mulc, 2019]. *Multilingual w/o LIDs:* We trained a multilingual Transformer TTS model using the paired data shown in Table 1 without language IDs (LIDs). *Multilingual w/ LIDs:* We trained a multilingual Transformer TTS model with the paired data of the unseen language. It also used the language IDs.

Unseen language. We compared *Fully zero-shot TTS* and *Text-seen zero-shot TTS* defined in § 2.3. In *Oracle*, we used the *Monolingual* and *Multilingual w/ LIDs*, which used the paired data of the unseen language. In *Fully zero-shot TTS*, we used *Multilingual w/o LIDs* to synthesize speech from text tokens in the unseen language. This method corresponds to the conventional multilingual TTS model using bytes [He *et al.*, 2021] or IPA symbols [Staib *et al.*, 2020].

Evaluation Metrics

To objectively measure the synthetic speech quality, we used mel cepstral distortion (MCD) [Fukada *et al.*, 1992] with the mel cepstrum dimension set to 25. We also evaluated the intelligibility using CERs computed with a multilingual ASR model [Radford *et al.*, 2022]. We used a pretrained *large* model that is publicly available[6]. To evaluate the naturalness, we carried out listening tests to calculate five-scale mean opinion scores (MOS) of synthesized speech for each method. Forty native speakers were recruited through Amazon Mechanical Turk [Paolacci *et al.*, 2010] for each of the tests. Furthermore, we leveraged a publicly available automatic MOS (AMOS) prediction model [Saeki *et al.*, 2022a] to evaluate the naturalness. Note that the model was trained on English and Chinese datasets, but previous work [Seki *et al.*, 2022] has reported that it also showed a correlation coefficient higher than 0.8 for another language (Japanese).

3.2 Evaluation Results on Seen Languages

We evaluated our framework on the seen languages included in the paired data, as defined in § 2.3. Table 2 lists the results in MCD and CER. Lower values are better for both metrics. As we can see, the byte-based or IPA-based models with the proposed multilingual pretraining performed the best across all languages and metrics. Among the baselines, byte-based monolingual and multilingual models tended to have higher MCD and CER than IPA-based models, and failed to synthesize intelligible speech in some languages. For example, the baseline byte-based models showed the high CER values for French, which has a deep orthography, meaning that a single character has different pronunciations depending on the context. We observed that our method improved the byte-based models and they outperformed the IPA-based baseline models for all the metrics and languages. It is worth noting that the proposed byte-based models even outperformed the proposed IPA-based models except for el and ru. These results suggest that our framework is effective in building a TTS model for languages without G2P modules.

3.3 Evaluation Results on Unseen Language

We evaluated our method on zero-shot TTS for the unseen language defined in § 2.3. As described in § 2.4, we first used the x-vector from the es speaker to compute the MCD. Table 3 lists the results. The baseline models showed the CERs of over 40% and MCDs of over 10.0. However, our

[5]The original paper [Li *et al.*, 2019b] also reports the instability.

[6]https://github.com/openai/whisper

Method	de		fr		ru		fi		hu		nl		el	
	MCD	CER	MCD	CER	MCD	CER	MCD	CER	MCD	CER	MCD	CER	MCD	CER
Natural	-	2.75	-	4.52	-	2.12	-	4.73	-	4.86	-	6.22	-	7.14
Baseline (Monolingual)														
Bytes monolingual	7.70	8.61	11.76	91.82	11.43	>100	8.33	56.03	10.22	93.05	7.49	15.33	10.20	85.98
IPA monolingual	7.38	4.07	8.96	17.86	11.89	25.30	7.23	27.62	7.59	24.62	7.80	19.20	8.16	21.79
Baseline (Multilingual)														
Bytes multilingual w/o LIDs	7.68	37.46	8.71	41.35	9.38	45.92	6.26	29.19	6.48	33.82	8.46	46.33	7.64	36.24
Bytes multilingual w/ LIDs	6.51	13.19	10.84	55.79	12.89	>100	6.78	27.22	9.09	42.97	8.47	39.37	7.25	23.56
IPA multilingual w/o LIDs	6.31	10.64	7.44	20.86	8.10	35.32	5.53	19.56	5.59	14.03	7.76	34.49	6.90	19.33
IPA multilingual w/ LIDs	6.16	9.76	6.88	14.97	7.63	23.54	5.17	10.63	5.28	9.11	6.95	19.48	6.90	16.97
Proposed (Unsupervised text pretraining)														
Bytes multilingual	**5.65**	**3.79**	**6.48**	**7.15**	7.38	**10.62**	**4.99**	**5.28**	**5.01**	**6.05**	**6.52**	**13.74**	6.57	11.75
IPA multilingual	5.88	5.52	6.61	7.72	**7.25**	15.85	5.18	8.62	5.30	7.37	7.00	14.42	**6.53**	**11.06**

Table 2: Evaluation results for *seen* languages. Bold indicates best scores in baseline and proposed methods.

Method	es		
	es x-vector		fr x-vector
	MCD	CER	CER
Natural	-	2.71	2.71
Oracle			
Bytes monolingual	8.65	10.70	-
IPA monolingual	8.47	5.28	-
IPA multilingual	6.20	5.32	6.99
Baseline (Fully zero-shot TTS)			
Bytes multilingual	11.22	64.07	66.45
IPA multilingual	10.75	44.75	44.37
Proposed (Text-seen zero-shot TTS)			
Bytes multilingual	9.05	18.27	13.74
IPA multilingual	9.44	11.69	13.33

Table 3: Evaluation results for *unseen* language.

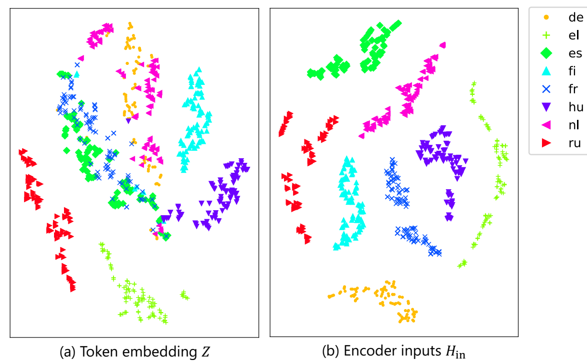

Figure 3: Visualization of token and language embedding. Pairs of similar languages (es–fr and de–nl) are overlapping in token embedding space, while output of bottleneck layer separates them.

proposed text preraining improved the metrics, resulting in CERs of less than half for both byte and IPA-based methods. Also, in contrast to the results for the seen languages, the IPA-based model outperformed the byte-based one in terms of CER. Compared with the oracle case with the paired data of the unseen language, our proposed zero-shot TTS showed higher MCD and CER but achieved only 1% difference in CER compared to the oracle byte-based monolingual model. These results demonstrate the effectiveness of our method in achieving intelligible zero-shot TTS for the unseen language.

To investigate the case where the target speaker information is completely unavailable, we also used the x-vector from a seen language. We chose the fr speaker because es and fr are both categorized as Western Romance in Glottolog [Hammarström *et al.*, 2021]. Table 3 lists the results. Note that this case does not have the MCD results, since a different speaker than the ground-truth speech was used. We can see that the unsupervised text pretraining also improved the zero-shot performance when using the x-vector from the fr speaker. In the proposed byte-based model, the cross-lingual x-vector showed the lower CER. This might result from that the es x-vector was not present in the training data whereas the fr x-vector was present in the training data.

3.4 Ablation Study

To further evaluate our method, we conducted several ablation studies. Table 4 lists the results. *Bytes multilingual* represents the byte-based proposed method in the evaluation of § 3.2 and 3.3. Note that it used the frozen language-aware embedding layer as formulated in Eq. (9). Some additional studies of our method are also presented in the Appendix.

In *W/o bottleneck layer*, we excluded the bottleneck layer and simply added the token and language embedding to obtain the encoder input in Eq. (2). We found that removing the bottleneck layer led to a performance drop in all the languages and metrics, with an average increase of 0.53 in MCD and 4.16% in CER. The largest increase was observed in the unseen language, with an increase of 1.21 in MCD. This suggests that the bottleneck layer, which projects the token and language embedding into the hidden input text representation with nonlinear dimensionality reduction, is effective in improving the generalization for zero-shot TTS.

We also evaluated the effect of including language IDs in the proposed method by comparing it with a version that excluded language IDs, referred to as *W/o language ID*. It corresponds to a simple multilingual BERT pretraining [Wu and Dredze, 2019] that uses only text tokens across different languages. We observed that the use of language IDs led to an

Method	Seen								Unseen		Avg.	
	de		fr		ru		fi		es			
	MCD	CER	MCD	CER	MCD	CER	MCD	CER	MCD	CER	MCD	CER
Bytes multilingual	5.65	3.79	6.48	7.15	7.38	10.62	4.99	5.28	9.05	18.27	6.46	**9.58**
W/o bottleneck layer	6.06	5.01	7.15	9.09	7.71	28.52	5.33	6.47	10.26	24.01	6.99	13.74
W/o language ID	6.07	5.09	7.09	9.99	7.77	22.58	5.23	6.99	10.45	32.70	6.96	14.06
W/o initializing encoder	5.59	3.75	6.52	9.31	7.12	16.47	4.86	5.03	9.02	21.91	**6.42**	11.85
Updating language-aware embedding layer	6.05	6.22	6.75	6.93	7.46	11.42	5.16	8.00	9.48	17.21	6.75	10.62

Table 4: Ablation studies on training and model configurations. Bold indicates best metrics on average (Avg.).

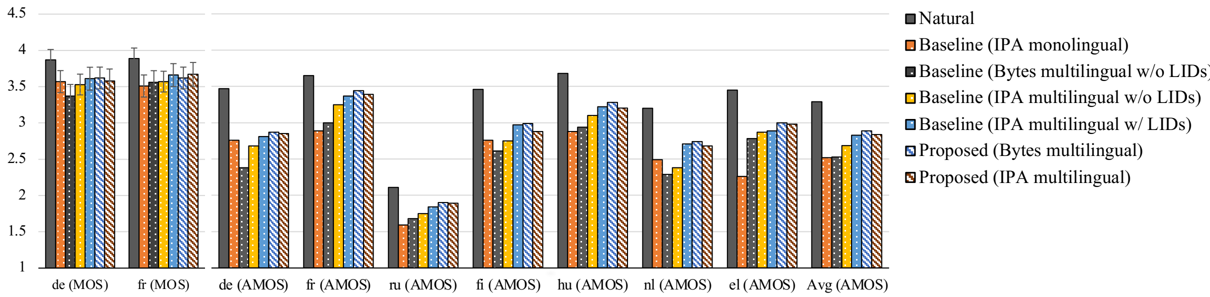

Figure 4: MOS and AMOS results for *seen* languages. Error bars in MOS results represent 95% confidence intervals.

Method	de		hu	
	MCD	CER	MCD	CER
Natural	-	2.75	-	2.12
Oracle				
IPA monolingual	7.38	4.07	7.59	24.62
IPA multilingual	6.16	9.76	5.28	9.11
Baseline (Fully zero-shot TTS)				
IPA multilingual	10.31	38.75	9.93	52.62
Proposed (Text-seen zero-shot TTS)				
Bytes multilingual	10.00	28.01	9.40	50.11

Table 5: Analysis on different *unseen* languages.

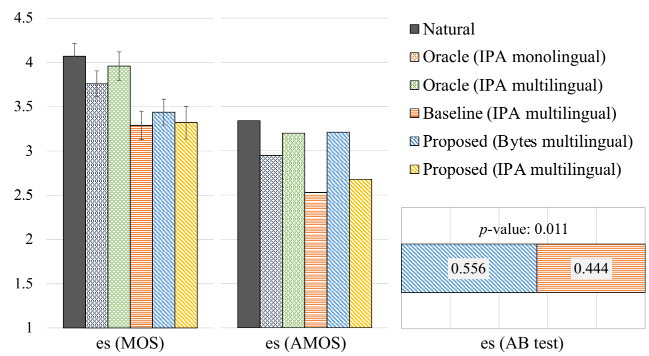

Figure 5: MOS, AMOS, and AB test results for *unseen* language. Error bars in MOS results represent 95% confidence intervals.

average improvement of 0.5 MCD and 4.48% CER, indicating the effectiveness of our approach in using language IDs.

In *W/o initializing encoder*, we did not initialize the encoder θ_{E} before the supervised leaning described in § 2.2. Instead, we only initialized the parameters θ_{T}, θ_{L}, and θ_{B} with the parameters pretrained in § 2.1. Through this evaluation, we investigated whether the performance gain with our method resulted from the initialization of the language-aware embedding layer or the encoder. We observed that *W/o initializing encoder* resulted in an improvement of 0.04 in MCD and only a 2.27% increase in CER on average, suggesting that our method benefits more from the pretraining of the language-aware embedding layer than from the encoder.

In *Updating language-aware embedding layer*, we updated the language-aware embedding layer during supervised learning, as formulated in Eq. (8). We observed that freezing the language-aware embedding layer led to better performance for most languages and metrics, resulting in an average difference of 0.29 in MCD and 1.04% in CER.

3.5 Dependency on Unseen Languages

We conducted evaluations on the zero-shot TTS for different unseen languages. The eight European languages included in the paired data are composed of Indo-European and Uralic language families defined in Glottolog [Hammarström *et al.*, 2021]. In this evaluation, we selected de and hu from each of the families. During supervised learning in § 2.2, we excluded the paired data for each of de and hu and instead included the paired data for es. Table 5 lists the results. We chose the IPA-based baseline method, which had shown better results in § 3.3. We observed that the pretraining improved the CER by around 10% and MCD by around 0.3 for de. However, the improvement in CER for hu was limited to 2%, while the MCD was improved by around 0.5. These results suggest that the performance of our zero-shot TTS is language dependent, as observed in previous work on cross-lingual transfer for NLP tasks [Wu and Dredze, 2019].

Fig. 3 visualize the token embedding Z and encoder inputs H_{in} averaged on each utterance. We used a t-distributed stochastic neighbor embeddings (t-SNE) [van der Maaten and Hinton, 2008]. We observed overlaps in the token embedding for (es, fr) and (de, nl), which are classified as Western Romance and West Germanic in Glottolog, respectively. The encoder inputs are separated in the embedding space for each language. The results in Table 5 and the visualization suggest that the cross-lingual transfer works better when similar languages sharing the token embedding space are present during supervised learning. However, for languages with distinct token and language embeddings, the cross-lingual transferability might be limited. We leave the further analysis on language dependencies as a topic for future research.

3.6 Subjective Evaluations on Naturalness

We conducted evaluations on naturalness as described in § 3.1. Fig. 4 shows the results for seen languages. Note that we conducted the listening tests for de and fr. For each language, either of the proposed methods showed the highest MOS, while we did not observe any significant difference between the proposed methods and the best baseline method, which was the IPA-based multilingual model with LIDs. To further validate our results, we also evaluated the naturalness with an AMOS prediction model, as shown in Fig. 4. We observed that the either of the proposed methods showed the highest scores in all the languages. On average, the byte-based and IPA-based proposed models showed 2.89 and 2.84, respectively, while the best baseline method obtained 2.83[7]. Additionally, we observed that the byte-based proposed model often scored higher than the IPA-based proposed models, which is consistent with the results in Table 2.

Fig. 5 shows the results for unseen languages. The oracle methods had the highest MOS of 3.76 and 3.96, and the baseline zero-shot method had the lowest MOS of 3.29. The proposed methods outperformed the baseline method, and the byte- and IPA-based models had the MOS of 3.44 and 3.32, respectively. The AMOS results were consistent with the listening test results, with the proposed zero-shot TTS methods outperforming the baseline method. In this evaluation, the proposed byte-based model scored 3.21 on the AMOS, while the oracle IPA-based model scored 3.20. To further validate the results, we conducted a preference AB test on naturalness with 25 rators. As shown in Fig. 5, our byte-based model significantly outperformed the baseline IPA-based model.

4 Related Work

Multilingual TTS. While previous work on multilingual TTS has primarily focused on resource-rich languages [Zen *et al.*, 2012; Li and Zen, 2016], there is growing interest in developing TTS models on low-resource languages. Several studies have explored the input tokens shared across languages such as bytes [Li *et al.*, 2019a; He *et al.*, 2021], IPA symbols [Gutkin, 2017], and articulatory features [Lux and Vu, 2022], to transfer knowledge from resource-rich to low-resource languages. Grapheme tokens can eliminate the per-language G2P knowledge, and previous work has built a byte-based TTS model for around 40 languages [He *et al.*, 2021]. There has been work using the phonological features derived from IPA to achieve the zero-shot TTS [Staib *et al.*, 2020]. Our framework achieves the zero-shot cross-lingual transfer with bytes by leveraging multilingual text pretraining. There have been studies on using untranscribed speech data for low-resource scenarios by leveraging VQ-VAE [Zhang and Lin, 2020] or an ASR model [Ren *et al.*, 2019; Ni *et al.*, 2022]. Other work [Saeki *et al.*, 2022b] has trained a massively multilingual TTS using paired TTS, paired ASR, unpaired speech, and unpaired text data. While it also performs text-only training as in our work, it still uses the paired speech-text data of the target languages. Our framework is simple and scalable, while pioneering a novel paradigm with the zero-shot TTS approach that relies only on text data.

Cross-lingual representation learning for NLP. There have been studies on learning cross-lingual representations that can be applied to various NLP tasks in different languages [Gouws *et al.*, 2015; Ruder *et al.*, 2019]. Recent work has highlighted the strong cross-lingual transferability of multilingual BERT [Devlin *et al.*, 2019], which has been observed to perform surprisingly well when transferred to other languages [Wu and Dredze, 2019; Conneau and Lample, 2019]. Building on this, our work leverages multilingual MLM pretraining for TTS, which improves byte-based TTS models without G2P knowledge and achieves zero-shot TTS.

Language model pretraining for TTS. Previous research has explored self-supervised text pretraining techniques for TTS. BERT models have been used to extract contextual embeddings and enhance the prosody of TTS [Hayashi *et al.*, 2019; Xu *et al.*, 2021]. Other studies have used phonemes jointly with graphemes [Jia *et al.*, 2021] or sub-phonemes [Zhang *et al.*, 2022] as the inputs of the MLM pretraining. Our work proposes multilingual MLM pretraining for TTS using text tokens shared across languages, rather than focusing on monolingual pretraining.

5 Conclusions

We presented a multilingual TTS framework that leverages unsupervised text pretraining. Our framework achieved highly intelligible zero-shot TTS for an unseen language, resulting in a CER of less than 12%. It also improved the TTS for seen languages, with byte-based models without G2P modules outperforming the IPA-based baselines. Our ablation studies provided additional insights, including the effectiveness of the frozen language embedding layer.

Limitations and future work. Our proposed framework has limitations. The performance gap remains between the oracle models and our zero-shot TTS models in terms of intelligibility, speech quality, and naturalness, as seen in the evaluation in § 3.3 and § 3.6. Further studies are needed to improve our zero-shot TTS. Our framework also has a limitation with language dependency, as the results in § 3.5 suggest that this dependency is caused by the presence of similar languages during supervised learning. Our future work will focus on studying this language dependency further and developing a method that performs better for various languages.

[7]The AMOS tended to be lower than the MOS. While the MOS prediction model has a high correlation, it may produce errors in predicting absolute values, as reported in previous work [Saeki *et al.*, 2022a]. The relative relationships are more reliable in the AMOS.

Acknowledgments

Part of this work was supported by JSPS KAKENHI Grant Number 21H05054, 22H03639, and 22J12040. This work used the Bridges system [Nystrom *et al.*, 2015], which is supported by NSF award number ACI-1445606, at the Pittsburgh Supercomputing Center. We would like to thank the research teams at Google through the internship of the first author for providing various insights on this topic.

References

[Ba *et al.*, 2016] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[Bapna *et al.*, 2019] Ankur Bapna, N. Arivazhagan, and Orhan Firat. Simple, scalable adaptation for neural machine translation. In *Proc. EMNLP-IJCNLP*, pages 1538–1548, 2019.

[Conneau and Lample, 2019] Alexis Conneau and Guillaume Lample. Cross-lingual language model pretraining. In *Proc. NeurIPS*, pages 7059–7069, 2019.

[Devlin *et al.*, 2019] Jacob Devlin, Ming-Wei Chang, Kenton Lee, et al. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proc. NAACL*, pages 4171–4186, 2019.

[Ebrahimi and Kann, 2021] Abteen Ebrahimi and Katharina Kann. How to adapt your pretrained multilingual model to 1600 languages. In *Proc. ACL-IJCNLP*, pages 4555–4567, 2021.

[Fukada *et al.*, 1992] Toshiaki Fukada, Keiichi Tokuda, Takao Kobayashi, et al. An adaptive algorithm for mel-cepstral analysis of speech. In *Proc. ICASSP*, pages 137–140, 1992.

[Gordon Jr, 2005] Raymond G. Gordon Jr. Ethnologue, languages of the world. https://www.ethnologue.com/, 2005. Accessed: 2023-05-27.

[Gouws *et al.*, 2015] Stephan Gouws, Yoshua Bengio, and Gregory S. Corrado. Bilbowa: Fast bilingual distributed representations without word alignments. In *Proc. ICML*, pages 748–756, 2015.

[Gutkin, 2017] Alexander Gutkin. Uniform multilingual multi-speaker acoustic model for statistical parametric speech synthesis of low-resourced languages. In *Proc. Interspeech*, pages 2183–2187, 2017.

[Hammarström *et al.*, 2021] Harald Hammarström, Robert Forkel, Martin Haspelmath, and Sebastian Bank. Glottolog 4.5. *Max Planck Institute for the Science of Human History*, 2021.

[Hayashi *et al.*, 2019] Tomoki Hayashi, Shinji Watanabe, Tomoki Toda, et al. Pre-trained text embeddings for enhanced text-to-speech synthesis. In *Proc. Interspeech*, pages 4430–4434, 2019.

[Hayashi *et al.*, 2021] Tomoki Hayashi, Ryuichi Yamamoto, Takenori Yoshimura, et al. Espnet2-tts: Extending the edge of tts research. *arXiv preprint arXiv:2110.07840*, 2021.

[He *et al.*, 2021] Mutian He, Jingzhou Yang, Lei He, et al. Multilingual byte2speech models for scalable low-resource speech synthesis. *arXiv preprint arXiv:2103.03541*, 2021.

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

[Jia *et al.*, 2021] Ye Jia, Heiga Zen, Jonathan Shen, et al. PnG BERT: Augmented BERT on phonemes and graphemes for neural TTS. *arXiv preprint arXiv:2103.15060*, 2021.

[Kim *et al.*, 2021] Jaehyeon Kim, Jungil Kong, and Juhee Son. Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech. In *Proc. ICML*, pages 5530–5540, 2021.

[Kong *et al.*, 2020] Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae. HiFi-GAN: Generative adversarial networks for efficient and high fidelity speech synthesis. *Proc. NeurIPS*, 33:17022–17033, 2020.

[Li and Zen, 2016] Bo Li and Heiga Zen. Multi-language multi-speaker acoustic modeling for LSTM-RNN based statistical parametric speech synthesis. In *Proc. Interspeech*, pages 2468–2472, 2016.

[Li *et al.*, 2019a] Bo Li, Yu Zhang, Tara Sainath, et al. Bytes are all you need: End-to-end multilingual speech recognition and synthesis with bytes. In *Proc. ICASSP*, pages 5621–5625, 2019.

[Li *et al.*, 2019b] Naihan Li, Shujie Liu, Yanqing Liu, et al. Neural speech synthesis with Transformer network. In *Proc. AAAI*, pages 6706–6713, 2019.

[Li *et al.*, 2022] Xinjian Li, Florian Metze, David R. Mortensen, et al. ASR2K: Speech recognition for around 2000 languages without audio. *arXiv preprint arXiv:2209.02842*, 2022.

[Lux and Vu, 2022] Florian Lux and Ngoc Thang Vu. Language-agnostic meta-learning for low-resource text-to-speech with articulatory features. In *Proc. ACL*, pages 6858–6868, 2022.

[Munich Artificial Intelligence Laboratories GmbH, 2017] Munich Artificial Intelligence Laboratories GmbH. The M-AILABS speech dataset. https://www.caito.de/2019/01/the-m-ailabs-speech-dataset/, 2017. Accessed: 2023-05-27.

[Nagrani *et al.*, 2017] Arsha Nagrani, Joon Son Chung, and Andrew Zisserman. VoxCeleb: A large-scale speaker identification dataset. In *Proc. Interspeech*, pages 2616–2620, 2017.

[Nair and Hinton, 2010] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proc. ICML*, 2010.

[Ni *et al.*, 2022] Junrui Ni, Liming Wang, Heting Gao, et al. Unsupervised text-to-speech synthesis by unsupervised automatic speech recognition. In *Proc. Interspeech*, pages 461–465, 2022.

[Nystrom *et al.*, 2015] Nicholas A. Nystrom, Michael J. Levine, Ralph Roskies, et al. Bridges: a uniquely flexible hpc resource for new communities and data analytics. In *Proc. XSEDE*, pages 1–8, 2015.

[Paolacci *et al.*, 2010] Gabriele Paolacci, Jesse Chandler, and Panagiotis G. Ipeirotis. Running experiments on amazon mechanical turk. *Judgment and Decision making*, 5(5):411–419, 2010.

[Park and Mulc, 2019] Kyubyong Park and Thomas Mulc. CSS10: A collection of single speaker speech datasets for 10 languages. *Proc. Interspeech*, pages 1566–1570, 2019.

[Pires *et al.*, 2019] Telmo Pires, Eva Schlinger, and Dan Garrette. How multilingual is multilingual BERT? In *Proc. ACL*, pages 4996–5001, 2019.

[Prakash *et al.*, 2019] Anusha Prakash, A Leela Thomas, S. Umesh, et al. Building multilingual end-to-end speech synthesisers for Indian languages. In *Proc. SSW*, pages 194–199, 2019.

[Radford *et al.*, 2022] Alec Radford, Jong Wook Kim, Tao Xu, et al. Robust speech recognition via large-scale weak supervision. *arXiv preprint arXiv:2212.04356*, 2022.

[Ravanelli *et al.*, 2021] Mirco Ravanelli, Titouan Parcollet, Peter William VanHarn Plantinga, et al. SpeechBrain: A general-purpose speech toolkit. *arXiv preprint arXiv:2106.04624*, 2021.

[Ren *et al.*, 2019] Yi Ren, Xu Tan, Tao Qin, et al. Almost unsupervised text to speech and automatic speech recognition. In *Proc. ICML*, pages 5410–5419, 2019.

[Ruder *et al.*, 2019] Sebastian Ruder, Ivan Vulic, and Anders Søgaard. A survey of cross-lingual word embedding models. *JAIR*, 65:569–631, 2019.

[Saeki *et al.*, 2022a] Takaaki Saeki, Detai Xin, Wataru Nakata, et al. UTMOS: UTokyo-SaruLab system for VoiceMOS Challenge 2022. In *Proc. Interspeech*, pages 4521–4525, 2022.

[Saeki *et al.*, 2022b] Takaaki Saeki, Heiga Zen, Zhehuai Chen, et al. Virtuoso: Massive multilingual speech-text joint semi-supervised learning for text-to-speech. *arXiv preprint arXiv:2210.15447*, 2022.

[Seki *et al.*, 2022] Kentaro Seki, Shinnosuke Takamichi, Takaaki Saeki, et al. Text-to-speech synthesis from dark data with evaluation-in-the-loop data selection. *arXiv preprint arXiv:2210.14850*, 2022.

[Shen *et al.*, 2018] Jonathan Shen, Ruoming Pang, Ron J Weiss, et al. Natural TTS synthesis by conditioning WaveNet on mel spectrogram predictions. In *Proc. ICASSP*, pages 4779–4783, 2018.

[Snyder *et al.*, 2018] David Snyder, Daniel Garcia-Romero, Gregory Sell, et al. X-vectors: Robust dnn embeddings for speaker recognition. In *Proc. ICASSP*, pages 5329–5333, 2018.

[Staib *et al.*, 2020] Marlene Staib, Tian Huey Teh, Alexandra Torresquintero, et al. Phonological features for 0-shot multilingual speech synthesis. In *Proc. Interspeech*, pages 2942–2946, 2020.

[Tachibana *et al.*, 2018] Hideyuki Tachibana, Katsuya Uenoyama, and Shunsuke Aihara. Efficiently trainable text-to-speech system based on deep convolutional networks with guided attention. In *Proc. ICASSP*, pages 4784–4788, 2018.

[van der Maaten and Hinton, 2008] Laurens van der Maaten and Geoffrey E. Hinton. Visualizing data using t-sne. *JMLR*, 9(11):2579–2605, 2008.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, et al. Attention is all you need. In *Proc. NeurIPS*, volume 30, 2017.

[Veaux *et al.*, 2017] Christophe Veaux, Junichi Yamagishi, Kirsten MacDonald, et al. CSTR VCTK corpus: English multi-speaker corpus for CSTR voice cloning toolkit. *University of Edinburgh. The Centre for Speech Technology Research (CSTR)*, 2017.

[Wang *et al.*, 2021] Changhan Wang, Morgane Riviere, Ann Lee, et al. VoxPopuli: A large-scale multilingual speech corpus for representation learning, semi-supervised learning and interpretation. In *Proc. ACL*, pages 993–1003, 2021.

[Watanabe *et al.*, 2018] Shinji Watanabe, Takaaki Hori, Shigeki Karita, et al. ESPnet: End-to-end speech processing toolkit. *Proc. Interspeech*, pages 2207–2211, 2018.

[Wu and Dredze, 2019] Shijie Wu and Mark Dredze. Beto, Bentz, Becas: The surprising cross-lingual effectiveness of BERT. In *Proc. EMNLP-IJCNLP*, pages 833–844, 2019.

[Xu *et al.*, 2021] Guanghui Xu, Wei Song, Zhengchen Zhang, et al. Improving prosody modelling with cross-utterance BERT embeddings for end-to-end speech synthesis. In *Proc. ICASSP*, pages 6079–6083, 2021.

[Zen *et al.*, 2012] Heiga Zen, Norbert Braunschweiler, Sabine Buchholz, et al. Statistical parametric speech synthesis based on speaker and language factorization. *TASLP*, 20(6):1713–1724, 2012.

[Zen *et al.*, 2019] Heiga Zen, Viet Dang, Rob Clark, et al. LibriTTS: A corpus derived from LibriSpeech for text-to-speech. In *Proc. Interspeech*, pages 1526–1530, 2019.

[Zhang and Lin, 2020] Haitong Zhang and Yue Lin. Unsupervised learning for sequence-to-sequence text-to-speech for low-resource languages. *Proc. Interspeech*, pages 3161–3165, 2020.

[Zhang *et al.*, 2022] Guangyan Zhang, Kaitao Song, Xu Tan, et al. Mixed-Phoneme BERT: Improving BERT with mixed phoneme and sup-phoneme representations for text to speech. In *Proc. Interspeech*, pages 456–460, 2022.