

VIRTUOSO: MASSIVE MULTILINGUAL SPEECH-TEXT JOINT SEMI-SUPERVISED LEARNING FOR TEXT-TO-SPEECH

Takaaki Saeki^{1,3*}, Heiga Zen¹, Zhehuai Chen², Nobuyuki Morioka¹, Gary Wang²,
Yu Zhang², Ankur Bapna², Andrew Rosenberg², Bhuvana Ramabhadran²

¹ Google, Japan ² Google, USA ³ The University of Tokyo, Japan

takaaki_saeki@ipc.i.u-tokyo.ac.jp, {heigazen,zhehuai}@google.com

ABSTRACT

This paper proposes *Virtuoso*, a massively multilingual speech-text joint semi-supervised learning framework for text-to-speech synthesis (TTS) models. Existing multilingual TTS typically supports tens of languages, which are a small fraction of the thousands of languages in the world. One difficulty to scale multilingual TTS to hundreds of languages is collecting high-quality speech-text paired data in low-resource languages. This study extends *Maestro*, a speech-text joint pretraining framework for automatic speech recognition (ASR), to speech generation tasks. To train a TTS model from various types of speech and text data, different training schemes are designed to handle supervised (paired TTS and ASR data) and unsupervised (untranscribed speech and unspoken text) datasets. Experimental evaluation shows that 1) multilingual TTS models trained on *Virtuoso* can achieve significantly better naturalness and intelligibility than baseline ones in seen languages, and 2) they can synthesize reasonably intelligible and naturally sounding speech for unseen languages where no high-quality paired TTS data is available.

Index Terms— Multilingual text-to-speech synthesis, massive multilingual pretraining, speech-text semi-supervised joint learning.

1. INTRODUCTION

With the remarkable progress of neural text-to-speech synthesis (TTS) methods, current multilingual TTS systems can synthesize human-like high-quality speech in multiple languages. Early work on multilingual TTS focused on building a TTS system for rich-resource languages. For example, Zen et al. [1] built a multilingual HMM-based statistical parametric speech synthesis (SPSS) from five Western European languages, and Li and Zen [2] developed a neural network-based multilingual SPSS from six Western European languages. Recently, the research community has started scaling multilingual TTS to tens of languages. He et al. [3] proposed a multilingual Byte2Speech TTS model, where 900-hour speech data of 43 languages was used. However, scaling it to hundreds of languages is still highly challenging due to the difficulty in collecting a large amount of high-quality paired TTS data for low-resource languages [3]. To cover thousands of languages, this paper aims to develop a technology that can scale multilingual TTS to hundreds of languages by using diverse speech and text data.

Semi-supervised and self-supervised learning has shown effectiveness for a wide range of speech and natural language processing tasks. Massive multilingual speech pretraining [4] has shown remarkable performance for downstream speech recognition tasks such as multilingual ASR and speech translation. Recently, it has been

extended to multimodal speech-text joint pretraining [5, 6] using speech-text pairs, untranscribed speech, and unspoken text. Although various approaches of massively multilingual self/semi-supervised learning have been attempted for speech recognition tasks, they have not been fully explored for multilingual speech generation tasks.

This paper proposes *Virtuoso*, a massive multilingual speech-text joint pretraining framework based on self-supervised and semi-supervised learning. It extends *Maestro* [6], a speech-text semi-supervised pretraining framework for ASR, to speech generation tasks. *Virtuoso* allows us to pretrain a multilingual TTS model using unsupervised (untranscribed speech and unspoken text) and supervised (paired TTS and ASR data) datasets with training schemes designed for them, which will allow the model to scale to hundreds of languages. This work has the following contributions:

- Proposing massive multilingual semi-supervised pretraining for TTS. It leverages different training schemes for “paired ASR”, “paired TTS”, “untranscribed speech” and “unspoken text” data, to train a single TTS model.
- Zero-shot TTS, where decent-quality TTS can be achieved for languages not included in the “paired TTS” data.

2. RELATED WORK

Large-scale self-/semi-supervised speech pretraining has been actively studied and applied to various downstream recognition tasks. In addition to speech-only pretraining [7–9], there are multimodal approaches such as TTS-based text injection [10] and speech-text joint pretraining [5, 11–13]. *Maestro* [6] performs the modality matching of speech and text embedding to learn speech-aware text representation and vice versa. *Virtuoso* extends *Maestro* to speech generation tasks by adding a speech decoder on *Maestro*’s shared encoder.

There have been prior studies on joint training of ASR and TTS to improve ASR [14, 15], to obtain alignments [16], and to scale ASR for low-resource settings [17, 18]. *Virtuoso* also jointly learns ASR and TTS models, where its shared encoder learns speech-text representation for both recognition and generation tasks.

While most of the existing studies on multilingual TTS [2, 19–22] have focused on a limited number of rich-resource languages, some studies have investigated low-resource languages [23, 24]. Some previous work has used a byte sequence [3, 25] as input text tokens to eliminate per-language modules for phoneme inputs and to learn linguistic representations shared across multiple languages. The prior work which is most similar to this paper is Byte2Speech [3], where a multilingual TTS model mapping a byte sequence to mel-spectrogram was trained from 900 hours of paired TTS data including 43 languages by 109 speakers. *Virtuoso* also uses graphemes or bytes

*This work was carried out as an intern at Google, Japan in 2022.

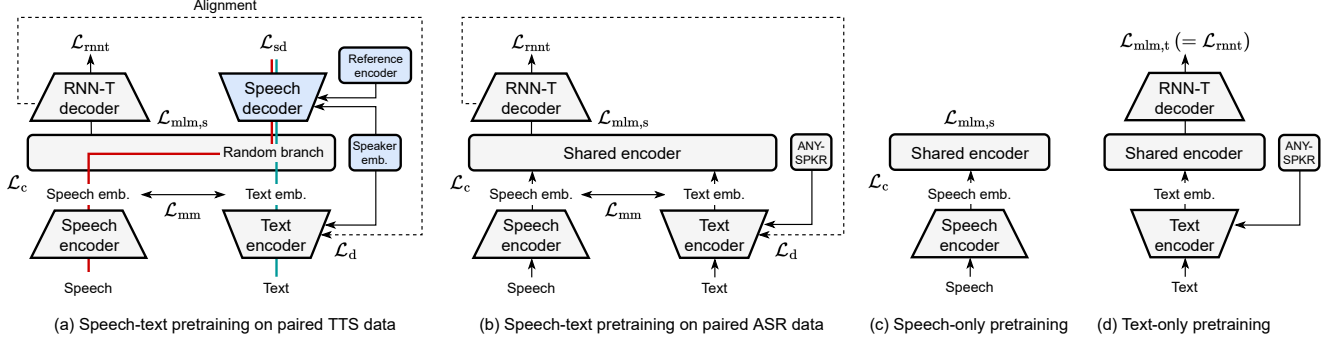


Fig. 1. Illustration of Virtuoso framework. (a) Whole architecture of Virtuoso and its supervised pretraining with paired TTS data where speech features are decoded with speaker labels and reference encoder output. (b) Supervised pretraining with paired ASR data. (c) Self-supervised pretraining with untranscribed speech. (d) Self-supervised pretraining with unspoken text. (b)–(d) follow those in Maestro.

as text input, whereas Virtuoso can use both paired and unpaired data by leveraging semi-supervised learning.

3. VIRTUOSO

3.1. Architecture

Fig. 1(a) shows the architecture of Virtuoso, consisting of a speech encoder, text encoder, shared encoder, RNN-T decoder, and speech decoder. It extends Maestro [6] by introducing the speech decoder to learn representation for speech generation tasks. The speech decoder predicts speech features with the decoder architecture used in Parallel Tacotron [26], given outputs from the shared encoder, a global reference encoder [27]), and speaker embedding.

Like Maestro, Virtuoso can use different types of text input, such as phonemes, graphemes, and bytes [6]¹. Although we can derive models for some downstream tasks from Virtuoso even without fine-tuning, e.g., TTS: text encoder→shared encoder→speech decoder, ASR: speech encoder→shared encoder→RNN-T decoder, and voice conversion: speech encoder→shared encoder→speech decoder, we can still fine-tune the model for these tasks.

3.2. Training objectives for supervised and unsupervised data

We aim to use both supervised (paired TTS and ASR) and unsupervised (untranscribed speech and unspoken text) data to train Virtuoso. The paired ASR data often contains background noise and channel distortions, whereas the paired TTS data is usually high-quality and recorded in a well-designed environment [28]. Also, speaker labels are often available for the paired TTS data, but not for other data types. To handle such differences in the data nature, Virtuoso uses different training objectives for each of them.

The training losses in Virtuoso include speech reconstruction loss \mathcal{L}_{sd} (to reconstruct speech features from shared representation), ASR loss \mathcal{L}_{rnt} [5, 6] (to decode text tokens from shared representation), contrastive loss \mathcal{L}_c and masked language modeling (MLM) loss $\mathcal{L}_{mlm,s}$ for speech embedding [8] (to learn self-supervised representation for speech), aligned MLM loss $\mathcal{L}_{mlm,t}$ for text embedding [6], duration loss \mathcal{L}_d [6] (to learn a token duration predictor), and modality matching (MM) loss \mathcal{L}_{mm} [6] (to unify speech and text embeddings). For \mathcal{L}_{sd} , we use the iterative loss [26], which computes the L_1 spectrogram loss at each of the lightweight convolutional blocks in the speech decoder.

¹In Maestro, the grapheme input showed the comparable performance to the phoneme, and the byte input improved the zero-resource performance.

3.2.1. Paired TTS data

Figure 1(a) also illustrates the training objective for the paired TTS data. The loss for the paired TTS data \mathcal{L}_{tts} is designed as

$$\mathcal{L}_{tts} = \lambda_{sd}\mathcal{L}_{sd} + \lambda_{rnt}\mathcal{L}_{rnt} + \lambda_c\mathcal{L}_c + \lambda_{mlm,s}\mathcal{L}_{mlm,s} + \lambda_d\mathcal{L}_d + \lambda_{mm}\mathcal{L}_{mm},$$

where λ denotes the weighting term of each objective. The speech decoder takes speaker embedding, a global reference encoder output, and shared encoder outputs then predicts speech features. Let the shared encoder output and speaker embedding be e_{shared} and e_{spkr} , respectively. The predicted speech features \hat{s} is given as $\hat{s} = \theta_{sd}(e_{shared}, e_{spkr}, \theta_{ref}(s))$, where θ_{ref} denotes the reference encoder and s is the target speech features.

The shared encoder inputs can be either speech embeddings from the speech encoder or up-sampled text embeddings from the text encoder. We denote the former as “speech branch” and the latter as “text branch”, where the former can be viewed as a masked autoencoder [29] and the latter is TTS. Although we can train the TTS model only with the text branch, a preliminary experiment showed that randomly switching between these branches during training helped the model to converge.

3.2.2. Paired ASR data

Figure 1(b) describes the training objective for the paired ASR data, which is identical to that in Maestro [6]. The objective for the ASR data \mathcal{L}_{asr} is given as

$$\mathcal{L}_{asr} = \lambda_{rnt}\mathcal{L}_{rnt} + \lambda_c\mathcal{L}_c + \lambda_{mlm,s}\mathcal{L}_{mlm,s} + \lambda_d\mathcal{L}_d + \lambda_{mm}\mathcal{L}_{mm}.$$

Note that we applied masking for speech and text embedding in the same manner as Maestro. As speaker labels are not available in the paired ASR data, \mathcal{L}_d is computed with a speaker embedding for the wildcard identifier ANY-SPKR.

3.2.3. Untranscribed speech and unspoken text data

Figures 1(c) and (d) depicts the objectives for the unsupervised data. For unsupervised data (untranscribed speech and unspoken text), we use the same self-supervised objectives as Maestro [6] as

$$\mathcal{L}_{speech-only} = \lambda_c\mathcal{L}_c + \lambda_{mlm,s}\mathcal{L}_{mlm,s}, \quad \mathcal{L}_{text-only} = \lambda_{mlm,t}\mathcal{L}_{mlm,t}.$$

As speaker labels are not available in unspoken text, upsampled text embedding is obtained using predicted durations with the speaker embedding for the wildcard identifier ANY-SPKR.

4. EXPERIMENTS

4.1. Experimental conditions

4.1.1. Dataset

A proprietary TTS dataset consisting of 1.5k hours of speech including 40 languages (English, Arabic, Mandarin, Czech, Danish, Spanish, Filipino, Gaelic, Hebrew, Hungarian, Icelandic, Javanese, Latvian, Norwegian, Dutch, Portuguese, Romanian, Russian, Slovakian, Slovenian, Ukrainian, Bengali, Welsh, German, Greek, Estonian, Farsi, Finnish, French, Hindi, Indonesian, Italian, Korean, Lithuanian, Malay, Polish, Serbian, Thai, and Vietnamese) was used as the paired TTS data. The total number of speakers in the paired TTS data was 284. The paired ASR data included VoxPopuli [30], MLS [31], Babel [32], and FLEURS [33] following Maestro-U [34]. The untranscribed speech data included 429k hours of speech-only data consisting of VoxPopuli, MLS, CommonVoice [35], and Babel as [6]. The unspoken text data contained the VoxPopuli text dataset (3GBytes) and mC4 [36] spanning 101 languages (15TBytes) [6].

4.1.2. Model specifications

The specifications of the RNN-T decoder and the speech encoder were the same as those of Maestro [6]. We used 10-layer Conformer blocks for the shared encoder. We concatenated the text token embedding with a 64-dimensional speaker embedding and then fed it to the text encoder. The other settings for the text encoder were the same as Maestro. The speech decoder had 8-headed self-attention blocks with lightweight convolutions, taking a sequence of 1,024-dimensional shared-encoder outputs, the speaker embedding, and the 8-dimensional global reference encoder output. The target of the speech decoder was a sequence of 80-dimensional mel-spectrogram extracted from a speech waveform at 16 kHz sampling (25 ms window length, 10 ms frame shift). We used UTF-8 bytes [25] for the byte inputs and 6,100 vocabulary size for the grapheme inputs. We used a WaveGrad neural vocoder with 50 iterations [37] to reconstruct speech waveform from a predicted mel-spectrogram.

4.1.3. Training specifications

We included the paired ASR, paired TTS, untranscribed speech, and unspoken text data with a fixed effective batch size of (256, 512, 1,024, 2,048) in each batch. We set the weighting terms ($\lambda_{sd}, \lambda_{rnt}, \lambda_c, \lambda_{mlm,s}, \lambda_d, \lambda_{mm}$) to (1.0, 4.0, 1.0, 1.0, 1.0, 0.3). While we set $\lambda_{mlm,t}$ to 2.0 without language ids, we upscaled it to 12.0 when injecting language ids as in Maestro-U [34]. We leveraged curriculum learning as in Maestro, while we started to include paired TTS and ASR data after 300k steps. We used the same learning rate scheduling and exponential moving average as Maestro. All the models described in Section 4.1.4 were trained for 500k iterations after starting to include the paired data. These models were trained for about two weeks using Google Cloud TPUs.

4.1.4. Models

We compared baseline models and Virtuoso with different input tokens and training data. Audio samples are available at [38]. In each method name, “G” and “B” indicate the grapheme and byte input, respectively. We trained two baseline models, (1) *Tacotron2-G-TTS*: Tacotron2 [39] trained on the paired TTS data with a grapheme sequence as input text representation, and (2) *MaestroFT-G-TTS*: Fine-tuned pretrained Maestro for the TTS task, where alignments between grapheme and speech features were computed using the Maestro’s pretrained speech encoder, shared encoder, and RNN-T

decoder then the text encoder and speech decoder were fine-tuned with the speech reconstruction loss.

We conducted an ablations study for Virtuoso with grapheme-based text representation. *Virtuoso-G-TTS* only used paired TTS data, *Virtuoso-G-Pair* used paired ASR and TTS data, *Virtuoso-G-All* used all the paired and unpaired data, and *Virtuoso-G-All-LID* introduced language IDs and a language adapter as in Maestro-U [34]. Finally, *Virtuoso-B-All-LID* used the UTF-8 bytes rather than graphemes as its text representation (like Byte2Speech [3]) with all the paired and unpaired data plus language IDs and a language adapter like *Virtuoso-G-All-LID*.

4.1.5. Evaluation metrics

We evaluated the models with three metrics. Since the models output speech features directly from graphemes or byte sequences, we used token error rates (TER) to evaluate the accuracy of linguistic content in synthetic speech. We used a multilingual ASR model trained on data that did not contain the paired TTS data but included all the languages in the evaluation. Subjective listening evaluations in naturalness using 5-scale mean opinion score (MOS) tests for three languages (English, Spanish, Tamil) were conducted to evaluate the synthetic speech. As it is difficult to have enough raters for some low-resource languages, we also used an automatically computed 5-scale MOS in naturalness by SQuID [40] (SQ). SQ doesn’t map perfectly to subjective MOS and is less sensitive to linguistic correctness since the model has largely seen ratings for high-quality TTS samples (ranging between 3.0 and 5.0). However, it is still useful for relative comparisons between models within the same language.

4.2. Experimental results

4.2.1. Seen languages

Among languages in the paired TTS data, we selected English, Spanish, Farsi, and Slovenian as “seen” languages for the evaluation. We selected one speaker for each language and used them in the evaluation. The left part of Table 1 shows the experimental results for seen languages². It can be seen from the TERs in the table that Virtuoso-G-TTS was significantly less intelligible than other models. This suggests that the paired TTS data was not enough to train a large-scale Virtuoso model. On the other hand, Virtuoso-G-Pair, which uses both TTS and ASR paired data, achieved better SQ and TER than the baseline models for all the seen languages. Introducing unsupervised data in addition to the supervised one had small or no impact both in SQ and TER. Among all models, Virtuoso-G-Pair consistently achieved the best SQ.

Table 2 also gives average TER and SQ over ten seen languages. Like the four seen languages in Table 1, Virtuoso-G-Pair achieved the highest SQ. As the Virtuoso-G-Pair models had more paired TTS data in a mini-batch than Virtuoso-G-All, the reconstruction loss could be smaller at the same number of training steps. The low SQ for Virtuoso-G-All-LID can be due to the larger loss weight for text-only data as in Maestro-U [34]. This could lead to the higher speech reconstruction loss values in the TTS learning. Further investigation and tuning of the training configurations is a future work.

4.2.2. Unseen languages

We selected Bulgarian, Afrikaans, Tamil, and Turkish as “unseen” languages. Note that the paired TTS data did not include these “unseen” languages whereas the paired ASR data and unsupervised

²TER for natural speech was worse than that of synthetic speech in Slovenian. This can be due to large variations in the natural speech.

Table 1. TER (%) and 5-scale SQ in naturalness for different languages. Smaller values are better for TERs whereas larger values are better for SQ. Values in the bold font indicate the best value.

	Seen languages								Unseen languages							
	English		Spanish		Farsi		Slovenian		Bulgarian		Afrikaans		Tamil		Turkish	
	TER	SQ	TER	SQ	TER	SQ	TER	SQ	TER	SQ	TER	SQ	TER	SQ	TER	SQ
Tacotron2-G-TTS	22.3	3.78	7.9	3.84	4.5	3.41	10.9	3.87	36.6	3.74	30.4	3.73	92.8	3.39	74.8	3.74
MaestroFT-G-TTS	19.1	3.74	7.6	4.00	5.6	3.66	13.9	3.87	30.0	3.81	24.1	3.68	95.2	2.62	81.9	3.99
Virtuoso-G-TTS	72.7	3.48	65.0	3.67	73.0	3.40	68.6	3.60	77.0	3.54	78.2	3.39	89.9	3.46	85.7	3.59
Virtuoso-G-Pair	16.6	4.01	7.1	4.06	4.9	3.85	6.8	3.99	23.5	3.83	25.6	4.02	27.4	4.35	38.0	4.02
Virtuoso-G-All	17.8	3.98	7.3	4.05	4.4	3.77	7.3	3.93	25.6	3.83	28.3	3.82	25.0	4.23	24.1	4.06
Virtuoso-G-All-LID	12.2	2.93	7.0	3.18	6.4	3.20	7.0	3.32	24.0	3.46	39.1	3.02	46.5	3.26	19.5	3.33
Virtuoso-B-All-LID	15.3	3.97	6.2	4.05	6.9	3.82	7.0	3.92	22.6	3.82	28.9	3.94	29.5	4.15	20.2	4.03
Natural	8.6	–	5.8	–	3.7	–	17.8	–	5.2	–	12.4	–	16.3	–	5.3	–

Table 2. Average TERs and SQ on 10 seen and 4 unseen languages.

	Seen		Unseen	
	TER	SQ	TER	SQ
Tacotron2-G-TTS	11.5	3.86	58.7	3.65
MaestroFT-G-TTS	11.0	3.93	57.8	3.53
Virtuoso-G-TTS	65.9	3.67	82.7	3.50
Virtuoso-G-Pair	9.9	4.05	28.6	4.06
Virtuoso-G-All	10.0	4.01	25.8	3.99
Virtuoso-G-All-LID	10.1	3.37	32.3	3.27
Virtuoso-B-All-LID	9.6	4.01	25.3	3.99
Natural	8.6	–	9.8	–

Table 3. Subjective 5-scale MOSs in naturalness by human raters. Values in the bold font indicate the best ones.

	English	Spanish	Tamil
Tacotron2-G-TTS	3.31±0.05	3.53±0.09	1.59±0.09
MaestroFT-G-TTS	3.67±0.04	3.66±0.07	1.24±0.05
Virtuoso-G-TTS	1.87±0.05	1.60±0.10	1.28±0.07
Virtuoso-G-Pair	3.79±0.04	3.96±0.07	3.39±0.08
Virtuoso-G-All	3.81±0.04	3.89±0.07	2.98±0.08
Virtuoso-G-All-LID	1.89±0.04	2.36±0.08	1.89±0.08
Virtuoso-B-All-LID	3.71±0.04	4.01±0.07	2.89±0.08

data included them (e.g., FLEURS [33], mC4 [36]). As there was no speaker for these languages in the paired TTS data, we selected one speaker from a similar seen language for each unseen language (Russian for Bulgarian, Dutch for Afrikaans, Hindi for Tamil, and French for Turkish) for evaluation. We found that using speaker embedding from a similar language gave better performance.

The right part of Table 1 shows the results for unseen languages. While the baseline models performed relatively well for Bulgarian and Afrikaans, they completely failed to synthesize intelligible speech for Tamil and Turkish. This can be because the input tokens for these two languages were significantly different from the seen languages in the paired TTS data. On the other hand, Virtuoso models with the paired ASR data achieved decent performance even for these unseen languages, as the additional paired ASR and unpaired data can provide some signals about these input tokens.

Table 2 shows that Virtuoso models with the unpaired data significantly improved TER over those without it, demonstrating that the

Table 4. The TER and SQ of the fine-tuned models using 1 hour (FT-1) and all (FT-All) of the paired TTS data for each language. Virtuoso-G-All was used as a base model for fine-tuning (Pretrain).

	Bulgarian		Afrikaans		Tamil		Turkish	
	TER	SQ	TER	SQ	TER	SQ	TER	SQ
Pretrain	25.6	3.83	28.3	3.82	25.0	4.23	24.1	4.06
+ FT-1	11.0	4.06	16.2	3.87	18.7	4.28	8.3	3.94
+ FT-All	7.6	4.10	14.8	3.91	21.1	4.15	6.4	3.97
Natural	5.2	–	12.4	–	16.3	–	5.3	–

introduction of the unpaired data can improve the linguistic accuracy in unseen languages. Like the seen languages, Virtuoso-B-All-LID achieved the lowest TER, while Virtuoso-G-Pair got the highest SQ.

As SQ is less sensitive to linguistic correctness [40], we also conducted subjective 5-scale MOS by human raters. Table 3 lists the MOS test results in English (seen), Spanish (seen), and Tamil (unseen). Although there are inconsistency in absolute scores between Tables 1 and 3, their rankings are somewhat consistent between them. We can see that the best Virtuoso model showed encouraging MOS of 3.39 for the unseen language.

Finally, we conducted an experiment to fine-tune a pretrained Virtuoso-G-All model on unseen languages. During fine-tuning, all self-supervised losses were disabled; only ASR and reconstruction losses were used. Table 4 gives the experimental results. We can see that fine-tuning significantly improved TER even with 1 hour paired TTS data whereas SQ was less affected.

5. CONCLUSIONS

This paper presented *Virtuoso*, a massively multilingual joint speech-text semi-supervised learning framework for TTS. Multilingual TTS models can be trained using both supervised (paired ASR and TTS data) and unsupervised (untranscribed speech and unspoken text) data including hundreds of languages, by extending *Maestro* to synthetic tasks. Experimental results demonstrated that the multilingual TTS models achieved significantly more intelligible and natural synthetic speech than baseline grapheme-based Tacotron2 and fine-tuned *Maestro* models. We also demonstrated its capability to synthesize speech in languages without paired TTS data. It has a potential to greatly increase the language coverage in multilingual TTS using unpaired speech and text data.

Future work includes exploring more efficient ways to inject unpaired data and improving the quality. Training a multilingual model from the data in hundreds of languages is also future work.

6. REFERENCES

- [1] H. Zen, N. Braunschweiler, S. Buchholz, et al., “Statistical parametric speech synthesis based on speaker and language factorization,” *IEEE Trans. Audio Speech Lang. Process.*, vol. 20, no. 6, pp. 1713–1724, 2012.
- [2] B. Li and H. Zen, “Multi-language multi-speaker acoustic modeling for LSTM-RNN based statistical parametric speech synthesis,” in *Proc. Interspeech*, 2016.
- [3] M. He, J. Yang, L. He, et al., “Multilingual Byte2Speech models for scalable low-resource speech synthesis,” *arXiv:2103.03541*, 2021.
- [4] A. Conneau, A. Baevski, R. Collobert, et al., “Unsupervised cross-lingual representation learning for speech recognition,” in *Proc. Interspeech*, 2021, pp. 2426–2430.
- [5] A. Bapna, C. Cherry, Y. Zhang, et al., “mSLAM: Massively multilingual joint pre-training for speech and text,” *arXiv:2202.01374*, 2022.
- [6] Z. Chen, Y. Zhang, A. Rosenberg, et al., “MAESTRO: Matched speech text representations through modality matching,” in *Proc. Interspeech*, 2022, pp. 4093–4097.
- [7] A. Baevski, H. Zhou, A.-R. Mohamed, et al., “wav2vec 2.0: A framework for self-supervised learning of speech representations,” *arXiv:2006.11477*, 2020.
- [8] Y.-A. Chung, Y. Zhang, W. Han, et al., “w2v-BERT: Combining contrastive learning and masked language modeling for self-supervised speech pre-training,” *Proc. ASRU*, pp. 244–250, 2021.
- [9] S. Chen, C. Wang, Z. Chen, et al., “WavLM: Large-scale self-supervised pre-training for full stack speech processing,” *arXiv:2110.13900*, 2022.
- [10] Z. Chen, Y. Zhang, A. Rosenberg, et al., “Injecting text in self-supervised speech pretraining,” in *Proc. ASRU*, 2021, pp. 251–258.
- [11] Y. Tang, H. Gong, N. Dong, et al., “Unified speech-text pre-training for speech translation and recognition,” in *Proc. ACL*, 2022, pp. 1488–1499.
- [12] H. Bai, R. Zheng, J. Chen, et al., “A³T: Alignment-aware acoustic and text pretraining for speech synthesis and editing,” in *Proc. ICML*, 2022, pp. 1399–1411.
- [13] J. Ao, R. Wang, L. Zhou, et al., “SpeechT5: Unified-modal encoder-decoder pre-training for spoken language processing,” in *Proc. ACL*, 2022, pp. 5723–5738.
- [14] S. Karita, S. Watanabe, T. Iwata, et al., “Semi-supervised end-to-end speech recognition using text-to-speech and autoencoders,” in *Proc. ICASSP*, 2019, pp. 6166–6170.
- [15] G. Wang, A. Rosenberg, Z. Chen, et al., “Improving speech recognition using consistent predictions on synthesized speech,” *Proc. ICASSP*, pp. 7029–7033, 2020.
- [16] D. Lim, W. Jang, G. O, et al., “JDI-T: Jointly trained duration informed Transformer for text-to-speech without explicit alignment,” in *Proc. Interspeech*, 2020, pp. 4004–4008.
- [17] Y. Ren, X. Tan, T. Qin, et al., “Almost unsupervised text to speech and automatic speech recognition,” *arXiv:1905.06791*, 2019.
- [18] N. Makishima, S. Suzuki, A. Ando, et al., “Speaker consistency loss and step-wise optimization for semi-supervised joint training of TTS and ASR using unpaired text data,” in *Proc. Interspeech*, 2022, pp. 526–530.
- [19] D. O’Shaughnessy, “Multilingual text-to-speech synthesis: The Bell labs approach,” *Computational Linguistics*, vol. 24, no. 4, 1998.
- [20] Y. Zhang, R. J. Weiss, H. Zen, et al., “Learning to speak fluently in a foreign language: Multilingual speech synthesis and cross-language voice cloning,” in *Proc. Interspeech*, 2019, pp. 2080–2084.
- [21] S. Zhao, T. H. Nguyen, H. Wang, et al., “Towards natural bilingual and code-switched speech synthesis based on mix of monolingual recordings and cross-lingual voice conversion,” *Proc. Interspeech*, pp. 2927–2931, 2020.
- [22] M. Yang, S. Ding, T. Chen, et al., “Towards lifelong learning of multilingual text-to-speech synthesis,” in *Proc. ICASSP*, 2022, pp. 8022–8026.
- [23] A. Prakash, A. L. Thomas, S. Umesh, et al., “Building Multilingual End-to-End Speech Synthesizers for Indian Languages,” in *Proc. SSW*, 2019, pp. 194–199.
- [24] P. Ogayo, G. Neubig, and A. W. Black, “Building African voices,” in *Proc. Interspeech*, 2022, pp. 1263–1267.
- [25] B. Li, Y. Zhang, T. N. Sainath, et al., “Bytes are all you need: End-to-end multilingual speech recognition and synthesis with bytes,” *Proc. ICASSP*, pp. 5621–5625, 2019.
- [26] I. Elias, H. Zen, J. Shen, et al., “Parallel tacotron: Non-autoregressive and controllable tts,” in *Proc. ICASSP*, 2021, pp. 5709–5713.
- [27] W.-N. Hsu, Y. Zhang, R. Weiss, et al., “Hierarchical generative modeling for controllable speech synthesis,” in *Proc. ICLR*, 2019.
- [28] J. Yamagishi, C. Veaux, and K. MacDonald, “CSTR VCTK corpus: English multi-speaker corpus for CSTR voice cloning toolkit (version 0.92),” 2019.
- [29] K. He, X. Chen, S. Xie, et al., “Masked autoencoders are scalable vision learners,” in *Proc. CVPR*, 2022, pp. 15979–15988.
- [30] C. Wang, M. Riviere, A. Lee, et al., “VoxPopuli: A large-scale multilingual speech corpus for representation learning, semi-supervised learning and interpretation,” *arXiv:2101.00390*, 2021.
- [31] V. Pratap, Q. Xu, A. Sriram, et al., “MLS: A large-scale multilingual dataset for speech research,” *arXiv:2012.03411*, 2019.
- [32] M. J. Gales, K. M. Knill, A. Ragni, et al., “Speech recognition and keyword spotting for low-resource languages: Babel project research at CUED,” in *Proc. SLTU*, 2014, pp. 16–23.
- [33] A. Conneau, M. Ma, S. Khanuja, et al., “FLEURS: Few-shot learning evaluation of universal representations of speech,” *arXiv:2205.12446*, 2022.
- [34] Z. Chen, A. Bapna, A. Rosenberg, et al., “Maestro-U: leveraging joint speech-text representation learning for zero supervised speech ASR,” *arXiv:2210.10027*, 2022.
- [35] R. Ardila, M. Branson, K. Davis, et al., “Common Voice: A massively-multilingual speech corpus,” *arXiv:1912.06670*, 2019.
- [36] L. Xue, N. Constant, A. Roberts, et al., “mT5: A massively multilingual pre-trained text-to-text transformer,” *arXiv:2010.11934*, 2020.
- [37] N. Chen, Y. Zhang, H. Zen, et al., “WaveGrad: Estimating gradients for waveform generation,” in *Proc. ICLR*, 2021.
- [38] “Audio samples for Virtuoso,” <https://google.github.io/tacotron/publications/virtuoso/>.
- [39] J. Shen, R. Pang, R. J. Weiss, et al., “Natural TTS synthesis by conditioning WaveNet on mel spectrogram predictions,” in *Proc. ICASSP*, 2018, pp. 4779–4783.
- [40] T. Sellam, A. Bapna, J. Camp, et al., “SQuId: Measuring speech naturalness in many languages,” *arXiv:2210.06324*, 2022.