Rapid Speaker Adaptation in Low Resource Text to Speech Systems using Synthetic Data and Transfer learning

Raviraj Joshi
Flipkart, Bengaluru
raviraj.j@flipkart.com

Nikesh Garera
Flipkart, Bengaluru
nikesh.garera@flipkart.com

Abstract

Text-to-speech (TTS) systems are being built using end-to-end deep learning approaches. However, these systems require huge amounts of training data. We present our approach to built production quality TTS and perform speaker adaptation in extremely low resource settings. We propose a transfer learning approach using high-resource language data and synthetically generated data. We transfer the learnings from the out-domain high-resource English language. Further, we make use of out-of-the-box single-speaker TTS in the target language to generate in-domain synthetic data. We employ a three-step approach to train a high-quality single-speaker TTS system in a low-resource Indian language Hindi. We use a Tacotron2 like setup with a spectrogram prediction network and a waveglow vocoder. The Tacotron2 acoustic model is trained on English data, followed by synthetic Hindi data from the existing TTS system. Finally, the decoder of this model is fine-tuned on only 3 hours of target Hindi speaker data to enable rapid speaker adaptation. We show the importance of this dual pre-training and decoder-only fine-tuning using subjective MOS evaluation. Using transfer learning from high-resource language and synthetic corpus we present a low-cost solution to train a custom TTS model.

1 Introduction

Speech synthesis systems are widely used in applications like voice assistants and customer service voice bots (Joshi and Kannan, 2021; Joshi and Singh, 2022). They are used commonly along with automatic speech recognition (ASR) (Joshi and Kumar, 2022) systems to provide an end-to-end voice interface. Recently, text-to-speech (TTS) systems have been trained using end-to-end deep learning approaches (Shen et al., 2018). The TTS models are based on an independent acoustic model converting text to spectrogram and a vocoder converting spectrogram to speech. More recently, these two models have been integrated into the model directly converting text to target speech (Weiss et al., 2021). However, the single end-to-end model requires large amounts of transcribed data. The dual model approach can be trained on comparatively less data as training a vocoder only requires audio data without its text transcripts. In general, training an end-to-end TTS requires a large amount of high-quality studio recordings to build a production-quality model.

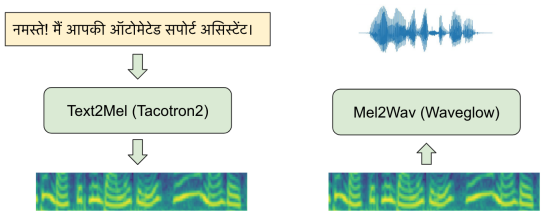

Figure 1: Overall flow of the TTS system

The popular text to spectrogram models include Tacotron2 (Shen et al., 2018), Transformer-TTS (Li et al., 2019), FastSpeech2 (Ren et al., 2020), FastPitch (Łańcucki, 2021), and Glow-TTS (Kim et al., 2020). In terms of voice quality the Tacotron2 model is still competitive with other models and less prone to over-fitting in low resource settings (Favaro et al., 2021; Abdelali et al., 2022; García et al., 2022; Finkelstein et al., 2022). There are multiple options for the vocoder as well like Clarinet (Ping et al., 2018), Waveglow (Prenger et al., 2019), MelGAN (Kumar et al., 2019), HiFiGAN (Kong et al., 2020), StyleMelGAN (Mustafa et al., 2021), and ParallelWaveGAN (Yamamoto et al., 2020). We choose Waveglow since it is competitive with other vocoders and is easy to train (Abdelali et al., 2022; García et al., 2022; Shih et al., 2021). There is other single model end-to-end architectures like VITS (Kim et al., 2021), Wave-Tacotron (Weiss et al., 2021) and JETS (Lim et al., 2022) for spectrogram-free TTS approaches. Although such models are more desirable since they remove

the vocoder spectrogram features mismatch during training and inference but do not work well in low-resource settings.

In this work, we explore the Tacotron2-based acoustic model and Waveglow-based vocoder to build a production-quality TTS system in low-resource, low-budget settings. The high-level flow is shown in Figure 1. In general, these models require 10 to 20 hours of quality data to train high-quality TTS systems. We aim to reduce the data requirements using simple strategies. Previous works in literature have proposed approaches to adapt to a new speaker with a few hours to a few minutes of data (Prakash and Murthy, 2020). However, these approaches have only been tested on some 10-30 utterances and might not be suitable for high-quality applications. Recently, a TTS system Vall-E (Wang et al., 2023) has shown extraordinary zero-shot capabilities. However, this system uses a complex architecture and requires 60K hours of pre-training data making it infeasible in low-resource scenarios.

In order to build low-resource TTS, we explore transfer learning from English data and synthetic audio corpus from the existing TTS model. We show the effectiveness of our approach in the context of the low-resource Indian language Hindi. While transfer learning from English is a common approach, we propose the usage of existing out-of-the-box TTS to further augment the data. Using an out-of-the-box TTS system has multiple advantages. It allows us to get a large amount of (real-text, synthetic-audio) pairs in the domain of our choice. It is a low-cost solution as compared to obtaining studio recordings for an equivalent amount of data. With high-quality out-of-the-box single-speaker text-to-speech systems available in the majority of languages, we leverage it to build a TTS in the speaker of our choice. We use an in-house single-speaker Hindi TTS system to generate synthetic corpus, however, the approach is applicable to any out-of-the-box TTS system. While we could have directly used the original training data of the initial TTS system, we utilized the model as a black box so that we can generate data in the domain of our choice.

We propose a three-step approach to build a low-cost TTS system. This approach is depicted in Figure 2.

- We initially pre-train the Tacotron2 acoustic model with public English LJSpeech data.
- We then ignore the initial character embedding layer based on English (Roman script) and re-train the entire model using a synthetic Hindi corpus (Devanagari script). This synthetic data pre-training step is important as it adapts the Tacotron2 encoder to the target domain text in the Devanagari script.
- Finally, we adapt the model to the target speaker using 3 hours of target speaker Hindi data. In the second step, although the audios are synthetic and from a different speaker, a large amount of real target domain text ensures high-quality pretraining of the text encoder. Therefore, during the final step, we freeze the encoder and only fine-tune the decoder of the Tacotron2 encoder-decoder model.

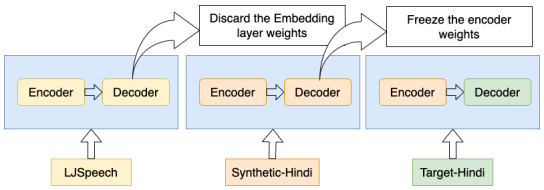

Figure 2: The training strategy for low resource TTS using less amount of Target-Hindi Data

We show that using this three-step strategy allows us to rapidly build a TTS system in the speaker of our choice. Although we can further reduce the data requirements overall stability of the model is impacted thus hindering its deployment in high-quality applications. We perform a subjective evaluation of our approach on an unseen domain with a larger test set and show its effectiveness.

2 Related Work

In this section, we describe the previous attempts to train a low-resource TTS system. A generic Indic TTS system using multiple languages and voices was built in (Prakash and Murthy, 2020). They used the tacotron2 + waveglow setup, along with a common linguistic representation multi-language character map (MLCM) to map all languages to a common script. They also utilize speaker embeddings based on x-vectors to train a multi-speaker model. Their observations indicate that such a system does not scale to the unseen speakers. So they propose a new speaker adaptation using only 7 minutes of data. However, the adapted system was tested only on 10 utterances which might not scale well to a larger set in high-quality settings.

In this work, we focus on the original script of the language as mapping it to a common script could result in information loss for some languages. Previously, a multi-lingual TTS system using MLCM character representation was introduced in (Prakash et al., 2019).

Cross-lingual transfer learning and data augmentation approach for low resource TTS were proposed in (Byambadorj et al., 2021). The spectrogram prediction network was trained using cross-lingual transfer learning (TL) from high resource language, data augmentation by varying parameters like pitch and speed, and a combination of two approaches. In the TL approach models were sequentially trained on high resource language (English) followed by a low-resource language (Mongolian). The input script was first converted into IPA phonetic format to enable the transfer of knowledge. We followed a similar approach in our first two steps but without using the common IPA phones instead of relying on the original script. This work also indicates a minimum of 3 hours of data is needed to cover all the phones. However, this approach is based on a multi-speaker TTS system as opposed to our single-speaker model. Similarly, data augmentation using a voice conversion module was explored in (Cai et al., 2023) and (Ribeiro et al., 2022).

Another approach for data augmentation using the parent TTS system was proposed in (Hwang et al., 2021). An auto-regressive TTS was first trained and used to generate large-scale synthetic corpora. This synthetic corpus along with real corpus is then used to train a non-auto-regressive TTS system. A similar approach utilizing synthetic corpus from existing TTS is explored in (Finkelstein et al., 2022; Song et al., 2022). Our work is similar to these approaches where the common aspect is to generate synthetic audio from another TTS system. However, these works train different TTS architectures for the same speaker and propose complicated training approaches. We instead focus on any-speaker out-of-the-box TTS system and propose a simple fine-tuning strategy.

Multi-speaker models leveraging external speaker embedding are commonly used to address speaker adaptation in low-resource settings. Transfer learning from external speaker verification models to Tacotron2 TTS was initially explored in (Jia et al., 2018). Further, zero-shot un-seen speaker adaptation using a similar speaker embedding approach was explored in (Cooper et al., 2020). Other approaches have been proposed over to time to combine speaker embedding and style embedding in Tacotron2 setup (Chung and Mak, 2021). In this work, we are only concerned about single-speaker models in this work and directly use the input script to preserve the original representation.

3 Model Architecture

We use a Tacotron2-based spectrogram prediction network followed by a Waveglow-based speech synthesis model. The two pass models are preferred in low resource settings as compared to fully end-to-end models. We observe that the vanilla Tacotron2 model is less prone to over-fitting in low-resource scenarios as compared to the vanilla Transformer based Tacotron model. Moreover, these models are competitive with more recent models in terms of audio quality and hence used to evaluate our transfer learning approaches (Tan et al., 2021). The approaches presented in this work are data-oriented and can be easily extended to any other model architecture.

3.1 Tacotron2

The Tacotron2 (Shen et al., 2018) model uses an auto-regressive spectrogram prediction network followed by a wavenet vocoder (van den Oord et al.). We describe the details of the spectrogram prediction network from Tacotron2 used in this work. The text-to-Mel spectrogram prediction is done using a sequence-to-sequence network. The input characters are converted into 512-dimensional embeddings and passed to the encoder-decoder network. The encoder consists of 3 stacked convolution layers with 512 filters and a filter size of 5 x 1. Each convolution layer is followed by batch normalization and relu activation. The convolution block is followed by a single Bi-LSTM layer with 512 units. The decoder is an auto-regressive network conditioned on encoder hidden representation. It uses location-sensitive attention (Chorowski et al., 2015) to compute the context vector during each time step. The decoder uses pre-net, containing 2 feed-forward layers (256 units) followed by relu units. The output of the pre-net is concatenated with the context vector and passed through two uni-LSTM layers with 1024 units. The output of lstm is again concatenated with the context vector and passed through a dense layer to predict the

spectrogram frame. At this step, another parallel projection predicts the stop token. The predicted frame is passed through a post-net comprising of 5 conv layers (512 filters, 5 x 1 filter size) each followed by batch norm and tanh non-linearity. The post-net predicts the residual to be added to spectrogram prediction to enhance the output. The mean squared error (MSE) is used as a loss function. The loss is computed on both output of LSTM projection and post-net projection.

The ground truth mel spectrogram is computed with STFT using 50 ms frame length and 12 ms hop. The STFT magnitude is transformed into a Mel scale using an 80-channel Mel filter bank. This is followed by log compression to get ground truth log-Mel spectrogram. Other hyperparameters are the same as those described in the original work.

3.2 Waveglow

Waveglow (Prenger et al., 2019) is a flow-based network that converts Mel-spectrogram to speech. It is a generative model that generates audio by sampling from a distribution. The samples are taken from zero mean, spherical Gaussian distribution, and transformed into audio distribution by passing it through a series of invertible transformations. It essentially models the audio distribution conditioned on a Mel-spectrogram. The model is trained by minimizing the log-likelihood of the data.

4 Dataset Details

We use three different datasets in this work. These datasets are single-speaker labeled audio-text pairs. One dataset is synthetically generated while the other two are real data. These are described below.

- **LJSpeech-English (24 hrs)**: It is a public domain single-speaker audio dataset consisting of 13,100 audio clips (Ito and Johnson, 2017). The text of the audio is taken from 7 non-fiction English books. The total length of the dataset is approximately 24 hours.

- **Syntethic-Hindi (15 hrs)**: A synthetic data set is created using an in-house TTS system. The output of the system was a single speaker and a female voice. Around 16k short utterances in Devanagari script majorly from the grocery voice assistant domain were converted to speech. The size of this dataset is around 15 hrs.

- **Real-Hindi (3 hrs)**: This is the target low-resource speaker data. A subset of utterances (2.5k) from the voice assistant domain were recorded in the voice of an external female speaker. These were high-quality studio recordings and the size of the dataset was around 3 hrs. We also evaluate the full 15 hrs of studio recordings of the above voice assistant utterances for comparative analysis.

All the audio data is re-sampled at 16kHz and encoded in 16-bit PCM wav format for training and inference.

5 Experimental Setup

We evaluate different variations of the pre-training strategy and try to come up with the best training strategy. We use English data and synthetic data for pre-training. First, the model is trained on English data followed by Hindi synthetic data. Finally, the model is trained on the target real Hindi corpus. The third corpus is the smallest in size and depicts the low-resource speaker. The final fine-tuning is done in two different ways. One approach is to perform full-finetuning and the second approach is to perform partial fine-tuning by freezing the encoder. The frozen encoder approach yields the best results and the corresponding flow is shown in Figure 2. The frozen text encoder is desirable since it is pre-trained on a large amount of target domain text as opposed to a small amount of data in the third step. Overall we consider the following pre-training strategies:

- **Direct target speaker (Hindi) training** - With just 3 hours of data and the results were mostly noisy and the training did not converge to a decent model. So the results of this model training are not discussed in the next sections.

- **LJSpeech English pre-training + Target-Hindi finetuning** - In this setup, since the input symbols of English and Hindi are completely different we discard the embedding layer weights and retain all other weights of the pre-trained English model. Post this we perform target Hindi data fine-tuning. Also, full fine-tuning is performed using target data since the embedding layer is part of the encoder and needs to be re-trained.

- **LJSpeech English pre-training + Synthetic Hindi pre-training + Target-Hindi full fine-tuning** - In this strategy, the model is initially

Training Strategy	MOS
Ground Truth Audios (Real)	4.65 ± 0.62
LJSpeech + Real (3 hrs)	4.27 ± 0.95
LJSpeech + Synthetic + Real (3 hrs)	4.54 ± 0.58
LJSpeech + Synthetic + Real (frozen encoder, 3 hrs)	$\mathbf{4.59} \pm 0.68$
LJSpeech + Synthetic + Real (frozen encoder, 15 hrs)	4.65 ± 0.58

Table 1: MOS scores for different training strategies

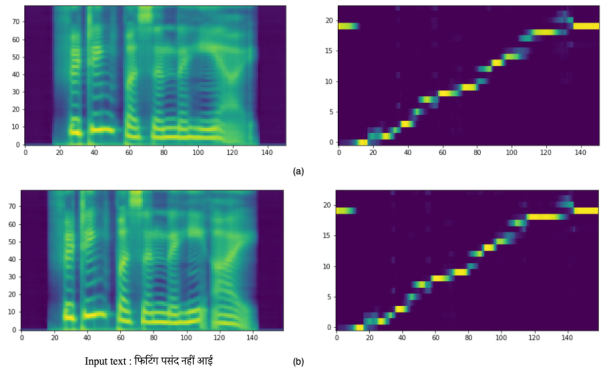

Figure 3: The Mel-spectrogram and attention alignment plot for a sample sentence using config (a) LJSpeech + Real (3 hrs) and (b) LJSpeech + Synthetic + Real (frozen encoder, 3 hrs). The difference in the resolution can be clearly seen at the end of the two spectrograms.

trained on English corpus, followed by full training on synthetic Hindi corpus. Finally, target Hindi speaker data is used to again fully fine-tune the models.

- **LJSpeech English pre-training + Synthetic Hindi pre-training + Target-Hindi decoder only finetuning** - This strategy is similar to the last strategy. The only difference is in the final fine-tuning only decoder weights are updated and the encoder is completely frozen. With this, we try to retain learnings from a much larger Hindi corpus.

6 Results and Discussion

We evaluate different pre-training strategies explored in this work using subjective mean opinion score (MOS) evaluation. A test set was created using 200 consumer experience (CX) voice bot interaction utterances. The domain of the test data (CX) is different from the domain of training utterances (voice assistant). This allows us to perform more rigorous testing of the model to unseen domains. These utterances were evaluated on (1-5) MOS scale by 10 specialized listeners with each audio evaluated by at least 3 listeners. The evaluators were explicitly trained for the evaluation activity and were part of the internal operations team thus ensuring high quality of evaluation. The audio were rated based on intelligibility and naturalness of the audios.

The results of the evaluation are shown in Table 1. The results indicate that LJSpeech English pre-training is helpful to create usable TTS models. Without this cross-lingual transfer learning, the model fails to produce intelligible output. Further training the model on synthetic Hindi data improves the output even further. Synthetic data pre-training is evaluated in two configurations. The full model is fine-tuned in the first config and the encoder is frozen in the second config. In both configurations, we perform LJSpeech pre-training. We observe that the frozen encoder approach yields superior performance in low-resource settings. All these experiments used 3 hrs of real Hindi data, 15 hours of synthetic Hindi data, and 24 hours of real English corpus. We also evaluate the three-step approach using full 15 hrs real data and see minimal improvements in performance. This indicates that 3 hrs of data is sufficient to build a production-ready TTS system with transfer learning from cross-lingual data and synthetic corpus. The plot of spectrogram and attention alignment weights for a sample sentence with and without using synthetic Hindi data is shown in Figure 3.

7 Conclusion

We present our transfer learning strategy to build low resource TTS system. We explore transfer learning from cross-lingual data and same-language synthetic data. The synthetic data is created using existing out of the box TTS system. The three-step approach involves pre-training with English data followed by synthetic Hindi data and low-resource real Hindi data. We evaluate these pre-training approaches using a strong out-of-domain test set using subjective MOS evaluation. In the

final step, we observe that the decoder only fine-tuning works better than full tuning. The high-level text representations (encoder output) of text trained on the large real text and synthetic audio pairs are better than just using the low-resource data.

References

Ahmed Abdelali, Nadir Durrani, Cenk Demiroglu, Fahim Dalvi, Hamdy Mubarak, and Kareem Darwish. 2022. Natiq: An end-to-end text-to-speech system for arabic. *arXiv preprint arXiv:2206.07373*.

Zolzaya Byambadorj, Ryota Nishimura, Altangerel Ayush, Kengo Ohta, and Norihide Kitaoka. 2021. Text-to-speech system for low-resource language using cross-lingual transfer learning and data augmentation. *EURASIP Journal on Audio, Speech, and Music Processing*, 2021(1):1–20.

Zexin Cai, Yaogen Yang, and Ming Li. 2023. Cross-lingual multi-speaker speech synthesis with limited bilingual training data. *Computer Speech & Language*, 77:101427.

Jan K Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio. 2015. Attention-based models for speech recognition. *Advances in neural information processing systems*, 28.

Raymond Chung and Brian Mak. 2021. On-the-fly data augmentation for text-to-speech style transfer. In *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 634–641. IEEE.

Erica Cooper, Cheng-I Lai, Yusuke Yasuda, Fuming Fang, Xin Wang, Nanxin Chen, and Junichi Yamagishi. 2020. Zero-shot multi-speaker text-to-speech with state-of-the-art neural speaker embeddings. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6184–6188. IEEE.

Anna Favaro, Licia Sbattella, Roberto Tedesco, and Vincenzo Scotti. 2021. Itacotron 2: transfering english speech synthesis architectures and speech features to italian. In *Proceedings of The Fourth International Conference on Natural Language and Speech Processing (ICNLSP 2021)*, pages 83–88.

Lev Finkelstein, Heiga Zen, Norman Casagrande, Chunan Chan, Ye Jia, Tom Kenter, Alexey Petelin, Jonathan Shen, Vincent Wan, Yu Zhang, et al. 2022. Training text-to-speech systems from synthetic data: A practical approach for accent transfer tasks. *arXiv preprint arXiv:2208.13183*.

Víctor García, Inma Hernáez, and Eva Navas. 2022. Evaluation of tacotron based synthesizers for spanish and basque. *Applied Sciences*, 12(3):1686.

Min-Jae Hwang, Ryuichi Yamamoto, Eunwoo Song, and Jae-Min Kim. 2021. Tts-by-tts: Tts-driven data augmentation for fast and high-quality speech synthesis. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6598–6602. IEEE.

Keith Ito and Linda Johnson. 2017. The lj speech dataset. https://keithito.com/LJ-Speech-Dataset/.

Ye Jia, Yu Zhang, Ron Weiss, Quan Wang, Jonathan Shen, Fei Ren, Patrick Nguyen, Ruoming Pang, Ignacio Lopez Moreno, Yonghui Wu, et al. 2018. Transfer learning from speaker verification to multispeaker text-to-speech synthesis. *Advances in neural information processing systems*, 31.

Raviraj Joshi and Venkateshan Kannan. 2021. Attention based end to end speech recognition for voice search in hindi and english. In *Proceedings of the 13th Annual Meeting of the Forum for Information Retrieval Evaluation*, pages 107–113.

Raviraj Joshi and Subodh Kumar. 2022. On comparison of encoders for attention based end to end speech recognition in standalone and rescoring mode. In *2022 IEEE International Conference on Signal Processing and Communications (SPCOM)*, pages 1–4. IEEE.

Raviraj Joshi and Anupam Singh. 2022. A simple baseline for domain adaptation in end to end asr systems using synthetic data. In *Proceedings of the Fifth Workshop on e-Commerce and NLP (ECNLP 5)*, pages 244–249.

Jaehyeon Kim, Sungwon Kim, Jungil Kong, and Sungroh Yoon. 2020. Glow-tts: A generative flow for text-to-speech via monotonic alignment search. *Advances in Neural Information Processing Systems*, 33:8067–8077.

Jaehyeon Kim, Jungil Kong, and Juhee Son. 2021. Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech. In *International Conference on Machine Learning*, pages 5530–5540. PMLR.

Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae. 2020. Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis. *Advances in Neural Information Processing Systems*, 33:17022–17033.

Kundan Kumar, Rithesh Kumar, Thibault de Boissiere, Lucas Gestin, Wei Zhen Teoh, Jose Sotelo, Alexandre de Brébisson, Yoshua Bengio, and Aaron C Courville. 2019. Melgan: Generative adversarial networks for conditional waveform synthesis. *Advances in neural information processing systems*, 32.

Adrian Łańcucki. 2021. Fastpitch: Parallel text-to-speech with pitch prediction. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6588–6592. IEEE.

Naihan Li, Shujie Liu, Yanqing Liu, Sheng Zhao, and Ming Liu. 2019. Neural speech synthesis with transformer network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6706–6713.

Dan Lim, Sunghee Jung, and Eesung Kim. 2022. Jets: Jointly training fastspeech2 and hifi-gan for end to end text to speech. *arXiv preprint arXiv:2203.16852*.

Ahmed Mustafa, Nicola Pia, and Guillaume Fuchs. 2021. Stylemelgan: An efficient high-fidelity adversarial vocoder with temporal adaptive normalization. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6034–6038. IEEE.

Wei Ping, Kainan Peng, and Jitong Chen. 2018. Clarinet: Parallel wave generation in end-to-end text-to-speech. *arXiv preprint arXiv:1807.07281*.

Anusha Prakash and Hema A Murthy. 2020. Generic indic text-to-speech synthesisers with rapid adaptation in an end-to-end framework. *Proc. Interspeech 2020*, pages 2962–2966.

Anusha Prakash, A Leela Thomas, S Umesh, and Hema A Murthy. 2019. Building multilingual end-to-end speech synthesisers for indian languages. In *Proc. of 10th ISCA Speech Synthesis Workshop (SSW'10)*, pages 194–199.

Ryan Prenger, Rafael Valle, and Bryan Catanzaro. 2019. Waveglow: A flow-based generative network for speech synthesis. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3617–3621. IEEE.

Yi Ren, Chenxu Hu, Xu Tan, Tao Qin, Sheng Zhao, Zhou Zhao, and Tie-Yan Liu. 2020. Fastspeech 2: Fast and high-quality end-to-end text to speech. *arXiv preprint arXiv:2006.04558*.

Manuel Sam Ribeiro, Julian Roth, Giulia Comini, Goeric Huybrechts, Adam Gabryś, and Jaime Lorenzo-Trueba. 2022. Cross-speaker style transfer for text-to-speech using data augmentation. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6797–6801. IEEE.

Jonathan Shen, Ruoming Pang, Ron J Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, Rj Skerrv-Ryan, et al. 2018. Natural tts synthesis by conditioning wavenet on mel spectrogram predictions. In *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 4779–4783. IEEE.

Kevin J Shih, Rafael Valle, Rohan Badlani, Adrian Lancucki, Wei Ping, and Bryan Catanzaro. 2021. 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*.

Eunwoo Song, Ryuichi Yamamoto, Ohsung Kwon, Chan-Ho Song, Min-Jae Hwang, Suhyeon Oh, Hyun-Wook Yoon, Jin-Seob Kim, and Jae-Min Kim. 2022. Tts-by-tts 2: Data-selective augmentation for neural speech synthesis using ranking support vector machine with variational autoencoder. *arXiv preprint arXiv:2206.14984*.

Xu Tan, Tao Qin, Frank Soong, and Tie-Yan Liu. 2021. A survey on neural speech synthesis. *arXiv preprint arXiv:2106.15561*.

Aäron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. In *9th ISCA Speech Synthesis Workshop*, pages 125–125.

Chengyi Wang, Sanyuan Chen, Yu Wu, Ziqiang Zhang, Long Zhou, Shujie Liu, Zhuo Chen, Yanqing Liu, Huaming Wang, Jinyu Li, et al. 2023. Neural codec language models are zero-shot text to speech synthesizers. *arXiv preprint arXiv:2301.02111*.

Ron J Weiss, RJ Skerry-Ryan, Eric Battenberg, Soroosh Mariooryad, and Diederik P Kingma. 2021. Wave-tacotron: Spectrogram-free end-to-end text-to-speech synthesis. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5679–5683. IEEE.

Ryuichi Yamamoto, Eunwoo Song, and Jae-Min Kim. 2020. Parallel wavegan: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6199–6203. IEEE.