TOWARDS BUILDING TEXT-TO-SPEECH SYSTEMS FOR THE NEXT BILLION USERS

Gokul Karthik Kumar[*†1,3,4] *Praveen S V*[*1,2]
Pratyush Kumar[1,2,4] *Mitesh M. Khapra*[1,2] *Karthik Nandakumar*[3]

[1]AI4Bharat, India
[2]Indian Institute of Technology Madras (IITM), India
[3]Mohamed Bin Zayed University of Artificial Intelligence (MBZUAI), UAE
[4]Microsoft Research, India

amalgamate = mixup

ABSTRACT

Deep learning based text-to-speech (TTS) systems have been evolving rapidly with advances in model architectures, training methodologies, and generalization across speakers and languages. However, these advances have not been thoroughly investigated for Indian language speech synthesis. Such investigation is computationally expensive given the number and diversity of Indian languages, relatively lower resource availability, and the diverse set of advances in neural TTS that remain untested. In this paper, we evaluate the choice of acoustic models, vocoders, supplementary loss functions, training schedules, and speaker and language diversity for Dravidian and Indo-Aryan languages. Based on this, we identify monolingual models with FastPitch and HiFi-GAN V1, trained jointly on male and female speakers to perform the best. With this setup, we train and evaluate TTS models for 13 languages and find our models to significantly improve upon existing models in all languages as measured by mean opinion scores. We open-source all models on the Bhashini platform [1].

Index Terms— text-to-speech, indian languages

1. INTRODUCTION

Deep neural networks have led to rapid progress in text-to-speech (TTS) systems. Compared to traditional methods like formant, concatenative, and statistical parametric speech synthesis, neural TTS achieves high-fidelity real-time speech synthesis with limited need for manual feature engineering [1]. This has enabled the generation of high-quality synthetic speech, which is being increasingly used in a larger number of applications.

A TTS system consists of 3 principal components: a text analysis module that converts text to linguistic features, an acoustic model that converts linguistic features to acoustic features, and a vocoder that converts acoustic features to speech waveforms. Many of the recent state-of-the-art TTS systems use two-stage speech synthesis models [2, 3, 4, 5], which amalgamate the first two components with an acoustic model to directly convert text to features such as spectrograms, or omit the text analysis module entirely. Within this class of models, several advances have been made. WaveNet [6] was one of the earliest works based on recurrent neural networks (RNNs) to generate speech waveforms directly from linguistic features. Tacotron [7] was the first successful neural acoustic model to generate spectrograms from the text directly. The use of an autoregressive TTS based on RNNs to generate speech waveforms was demonstrated in Tacotron2 [2]. The speed of the acoustic models was improved by replacing RNNs with Transformer-based non-autoregressive (NAR) acoustic models as demonstrated in FastSpeech [4] and FastPitch [5]. However, these NAR models require an external aligner module. The need for this aligner module was eliminated with the proposal of flow-based generative models, such as Glow-TTS [3], which implements a monotonic alignment search algorithm to map latent speech representations to representations in the text domain. Meanwhile, several neural vocoders like WaveGAN [8], MelGAN [9], HiFiGAN [10] and Multi-Band MelGAN [11] adapted Generative Adversarial Networks (GANs) for generating audio waveforms, which improve quality of generated speech, primarily with changes in the discriminator and addition of new loss functions. Recently, neural speech synthesizers [12, 13] based on denoising probabilistic diffusion have been proposed which generate high-quality speech but tend to be slower in inference owing to their iterative nature. While two-stage TTS systems remain popular, there is an ongoing exploration of end-to-end systems, such as VITS [14], that directly synthesizes speech from text.

Apart from advances in neural architectures, there has been an interest in developing TTS systems for low-resource settings. One approach is to study multi-speaker generalization. This has been studied for English [3, 5, 14], with models that can generate speech for multiple speakers as represented by speaker embeddings. Such models also have the practical benefit of efficient deployment in supporting multiple voices (say, one male and one female) from a single hosted model. Another approach is to consider multilingual generalization [15] to transfer knowledge from high resource languages by mapping the embeddings of the phoneme sets from different languages. Recently, YourTTS [16] successfully extended the end-to-end VITS model for multilingual generalization by conditioning on language embeddings.

The above paragraphs briefly summarize the advances in neural TTS over half a decade of active research. A characteristic of TTS, somewhat different from other domains such as computer vision, language modelling, neural translation, and speech recognition,

*Equal contribution
†Work started with internship at Microsoft and concluded at MBZUAI
[1]https://bhashini.gov.in/ulca/model/explore-models

NAR = non auto regressive

is that there is a large diversity of neural architectures and modelling techniques that continue to remain competitive. In other words, there is no one dominant TTS design methodology that is conclusively superior. Thus, the task of bringing the latest advances in TTS research to a set of languages requires that various design methodologies be implemented and tested with human evaluation. This is particularly challenging for Indian languages which are not only numerous but also significantly differ in terms of phonetics, morphology, word semantics, syntax and written scripts.

There have been a few studies specifically focused on Indian languages. For example, Vakyansh [17] open-sourced TTS models for 9 Indian languages with a combination of Glow-TTS with HiFi-GAN. Similarly, multilingual TTS models for Indian languages within the same family have been built [18] using Tacotron2 with WaveGlow [19] by making use of the multi-lingual character map [20] and the common label set [21]. However, various recent advances in TTS systems remain to be tested for Indian languages. For instance, TTS systems have not been built and evaluated for Indian languages that exhibit the following: fast generation as in FastPitch model [5], flow-based generation as in GlowTTS model [3], end-to-end generation as in VITS model [14], comparison of GAN based vocoders as in HiFiGAN [10] and Multi-Band MelGAN[11] with conditional models for waveform generation as in WaveGrad [22]. Also, the efficacy of multi-speaker models for Indian languages remains untested. Finally, multi-lingual models that group languages by their language family need to be explored for the latest state-of-the-art models.

In this paper, we aim to partially address this gap with a rigorous exploration of various TTS systems for 13 Indian languages across choices of acoustic models, vocoders, supplementary loss functions, training schedules, and speaker and language generation. Specifically, we consider three different acoustic models - FastPitch [5], GlowTTS [3], and the end-to-end VITS [14], three different vocoders - HiFiGAN V1 [10], Multi-Band MelGAN [11], and WaveGrad [22], use of SSIM [23] based supplementary loss, multi-speaker training with male and female voices, and multilingual models for four Dravidian, seven Indo-Aryan, and two Sino-Tibetan languages. With manual MOS-based and automated metric-based evaluation, we identify the combination of FastPitch and HiFiGAN V1, trained with male and female speakers, for a single language to be the preferred setup. With this setup, we train TTS models for the 13 languages and establish that these models improve upon existing TTS models with both MOS and automated metrics.

The real-world impact of natural sounding TTS is significant in a country like India with the need to deliver digital touch-points to a large population speaking 122 major languages across 5 different language families, with 25% of the population with reading disabilities. Thus, there is value for "foundation models" [24] in TTS to be available in the open-source to enable rapid innovation and deployment. This movement towards open-sourcing has been gaining momentum in other areas such as language modelling [25, 26], transliteration [27], translation [28], and speech recognition [29]. To contribute towards this effort, we open-source all our TTS models through the Bhashini platform [30].

2. DESIGN CHOICES FOR TTS MODELS

In this section, we detail the different model architectures, training strategies, and generalization techniques that we evaluate for TTS models for Indian languages.

2.1. Acoustic models

FastPitch[5]: To represent fast NAR models and acoustic models based on Transformers, we consider FastPitch [5]. The model is based on the feed-forward Transformer consisting of an encoder, 1-D convolution based duration and pitch predictors, and a decoder. The pitch and duration predictors are trained using mean-squared-error losses, but unlike [5] we extract ground truth frequencies from WORLD [31] and train the duration predictor on durations learnt from an alignment learning framework [32].
Glow-TTS [3]: To represent flow-based generative models we consider Glow-TTS. Similar to FastPitch, Glow-TTS uses a Transformer based encoder with slight modifications [3]. The model eliminates the need for an external aligner and uses the default Monotonic Alignment Search (MAS) algorithm trained with maximum likelihood estimation. The decoder is composed of a family of invertible flows and transforms a prior distribution into mel-spectrograms.
VITS [14]: To represent end-to-end models, we consider VITS [3]. The model uses the same text-encoder as Glow-TTS [3], a posterior encoder consisting of non-causal residual WaveNet blocks, a decoder based on HiFiGAN-V1 [10] a multi-period discriminator and a stochastic duration predictor.

We did not consider autoregressive methods such as Tacotron2 as Glow-TTS, Fast Pitch and VITS have been demonstrated to be better. However, as a limitation of our work, we do not include the most recent diffusion-based acoustic models such as Grad-TTS [13].

2.2. Vocoders

For the choice of vocoders, we restrict ourselves to those that use mel-spectrograms as the input representation. We do not consider auto-regressive models such as WaveNet [6] which have been improved upon by other vocoders both on generation speed and quality. In this work, we consider the following vocoders -
HiFi-GAN V1 [10]: The neural vocoder is based on a generative adversarial network and achieves high computational efficiency and sample quality. By using a single generator and multi-scale and multi-period discriminators, operating on different scales and periods of the input waveform, it captures implicit structures and long-term dependencies and is able to efficiently generate high-fidelity speech.
Multi-Band MelGAN [11]: This neural vocoder extends MelGAN [9] by doubling the receptive field for improved speech generation. It also substitutes the feature matching loss in MelGAN [9] with a multi-resolution STFT loss evaluated at both sub-band and full-band scales. Further, by adopting shared parameters for all sub-band signal predictions, Multi-Band MelGAN is able to achieve better speech with faster generation speeds.
WaveGrad [22]: We also consider a diffusion-based vocoder, namely WaveGrad [22], which is a non-autoregressive neural vocoder that iteratively transforms white Gaussian noise into high-quality audio waveforms by using a gradient-based sampler conditioned on mel-spectrograms.
As a limitation of our work, we do not consider flow-based vocoders such as WaveGlow. [19].

2.3. Training Strategies

Different TTS works employ different loss functions to enhance training, such as revisiting SSIM loss [33] or employing an ASR-based speech consistency loss [34] to enhance training. In this work, we consider two supplementary loss functions - the SSIM loss on the synthesized mel-spectrogram and a speech consistency loss [23]

that characterizes the intelligibility of the generated speech. The SSIM loss measures the structural similarity between the synthesized mel-spectrogram and ground truth mel-spectrogram on three dimensions: luminance, contrast and structure. The ASR loss measures the l_1-norm between convolutional features of the ground truth and synthesized mel-spectrograms extracted from the intermediate layers of the pretrained joint CTC-attention VGG-BLSTM network [34]. In addition, we consider training schedules where the alignment loss is turned off after a fixed number of steps.

2.4. Multi-speaker models

When building multi-speaker models, a common approach is to introduce speaker embeddings to condition the acoustic models. There are two ways of computing speaker embeddings: learn the speaker embedding network while training the acoustic model [35, 36] and use an external pretrained speaker verification model [37] to precompute embeddings. Since pretrained speaker verification models for Indian languages are not easily available, we use the former approach. We learn speaker embeddings and perform a point-wise vector addition with the encoder output of the acoustic model.

2.5. Multilingual models

To train multilingual models, we condition the text encoder outputs on language IDs using learnable embeddings optimized during training. All our acoustic models take in the raw text as input, instead of phonemes. To aid the generalization of the representations computed by the text encoder, we choose to map the diverse scripts of different Indian languages into a common representation. We do this by transliterating all scripts to the ISO format with the help of Aksharamukha [2]. As future work, we would also like to explore the common label set for Indian languages proposed in [21].

3. EXPERIMENTS

3.1. Experimental setup

Dataset: We use the latest version of the IndicTTS Database [38] with over 272 hours of transcribed speech recordings for 13 Indian languages including Assamese, Bengali, Bodo, Gujarati, Hindi, Kannada, Malayalam, Manipuri, Marathi, Odia, Rajasthani, Tamil and Telugu. All languages except Bodo have both male and female speakers, while Bodo only has a female speaker. For each speaker, there exists at least 8 hours of transcribed data. All audio samples are downsampled to a sampling rate of 22.05KHz and utterances having a duration greater than 20 seconds are filtered out. We preprocess text by replacing semi-colons and colons with commas, removing parenthesis symbols and collapsing whitespaces.

Training & Inference: We implement our models using Coqui-TTS[3] library. All models are trained for 2500 epochs on a single NVIDIA A100 40GB Tensor Core GPU, with a batch size of 32 and the default learning rate scheduling of Coqui-TTS. We train FastPitch with Adam optimizer with $\beta_1 = 0.99$ and $\beta_2 = 0.998$ with weight decay of $\lambda = 10^{-6}$. Glow-TTS is trained with RAdam optimizer with $\beta_1 = 0.99$ and $\beta_2 = 0.998$ with weight decay of $\lambda = 10^{-6}$. VITS is trained using the AdamW optimizer with $\beta_1 = 0.8$ and $\beta_2 = 0.99$ with weight decay of $\lambda = 0.01$. Each model took approximately 3 days to train. For our final models in Section 3.6, we turned off the aligner for the last 1000 epochs, as this helped us achieve better convergence in spectrogram reconstruction. While training vocoders, we use the default hyper-parameter settings in Coqui-TTS, except for WaveGrad where we increase the batch size to 96. We train a separate vocoder for each language. We observed that having large variations in average utterance durations between individual speakers for a given language make it difficult for multi-speaker acoustic models to learn alignments between input text and mel-spectrograms. For example, while training a multi-speaker model for the Telugu speaker, where the average utterance duration for the female speaker was nearly twice that of the male speaker, we observed the alignment loss failed to converge. This was resolved by modulating the tempo of the Telugu female's utterances to be 0.77x its original speed. Multilingual models for each language family are trained with the help of the Aksharamukha tool that supports transliteration for 10 of 13 Indian languages excluding Bodo, Manipuri, and Rajasthani. We post-process the generated audio samples with DCCRN [39] speech enhancement model to remove background artefacts.

Evaluation: We evaluate our models using subjective and objective metrics on a validation set of 30 utterances unseen during training. We conduct a subjective Mean Opinion Score (MOS) evaluation on LabelStudio [40] with the help of 42 raters, all of whom are native speakers of the language they are tasked to evaluate. This includes 6 raters each for Tamil and Hindi, 1 rater for Rajasthani and 3 raters for each of the remaining 10 languages. In Figure 1, we depict the annotation interface, where evaluators are requested to rate each audio sample on a scale of five [1 - Bad, 2 - Poor, 3 - Fair, 4 - Good and 5 - Excellent].

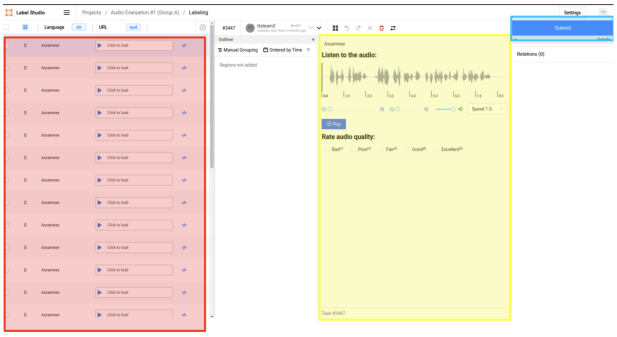

Fig. 1. Annotation Interface for MOS Evaluation on Label Studio. Raters use the red-shaded pane to navigate across audio samples, the yellow-shaded pane to rate the sample and submit their final ratings using the blue submit button.

To measure acoustics objectively, we use two metrics: mel-cepstral distortion (MCD) [41] and root-mean-square error of the log of the fundamental frequencies (F_0), with dynamic time warping [42] to temporally align the sequences. To measure intelligibility objectively, we use the character error rate (CER) with text extracted from Google Cloud's Automatic Speech Recognition[4]. Dravidian and Indo-Aryan languages have very distinct characteristics [18]. As it is computationally expensive to experiment with all 13 Indian languages, we choose one language under each family: Tamil (Dravidian) and Hindi (Indo-Aryan) to evaluate the design choices for TTS models.

[2] https://aksharamukha.appspot.com/
[3] https://github.com/coqui-ai/TTS

Metrics	MOS				MCD			F_0			CER			
Language	**GT**	**Ours**	**D**	**V**	**Ours**	**D**	**V**	**Ours**	**D**	**V**	**GT**	**Ours**	**D**	**V**
Dravidian Languages														
Kannada	4.11	**3.68**	2.90	2.27	**8.47**	13.11	9.64	**0.27**	0.35	0.30	0.132	**0.120**	0.217	0.391
Malayalam	4.24	**3.64**	3.49	1.92	**8.08**	12.48	9.31	**0.18**	0.21	0.22	0.123	**0.216**	0.303	0.337
Tamil	4.16	**3.84**	3.01	2.59	10.57	11.92	**7.28**	**0.31**	0.38	0.35	0.108	**0.107**	0.245	0.134
Telugu	4.42	**3.66**	3.40	2.61	9.91	11.03	**7.16**	**0.36**	0.38	0.37	0.273	**0.266**	2.254	0.364
Indo-Aryan Languages														
Assamese	3.63	**2.39**	-	-	**10.93**	-	-	**0.38**	-	-	-	-	-	-
Bengali	4.58	**3.37**	-	3.16	10.18	-	**5.97**	**0.28**	-	**0.28**	0.145	**0.167**	-	0.193
Gujarati	4.12	**3.58**	-	3.02	8.99	-	**5.56**	0.38	-	**0.36**	0.156	0.194	-	**0.182**
Hindi	4.33	**4.00**	3.70	3.16	8.73	12.41	**8.50**	**0.21**	0.23	0.24	0.104	**0.094**	0.127	0.100
Marathi	4.30	**3.26**	2.69	2.61	**9.51**	14.84	11.53	**0.30**	0.41	0.50	0.093	**0.075**	0.136	1.484
Odia	4.77	**4.19**	3.04	3.56	**8.90**	15.72	11.07	**0.22**	0.33	0.24	-	-	-	-
Rajasthani	4.10	**3.40**	3.37	-	**9.72**	12.89	-	**0.25**	0.30	-	-	-	-	-
Sino-Tibetan Languages														
Bodo	4.53	**3.53**	-	-	**9.89**	-	-	**0.23**	-	-	-	-	-	-
Manipuri	4.58	**3.30**	-	-	**11.68**	-	-	**0.27**	-	-	-	-	-	-

Table 1. Results of our model and existing works on the IndicTTS Database in terms of acoustic metrics (MCD, F_0), intelligibility (CER) and subjective scores (MOS) for evaluating naturalness of generated samples. GT: Ground Truth, Ours: AI4Bharat-TTS FastPitch+HiFiGAN, D: DON Lab's Tacotron2+WaveGlow [18], V: Vakyansh's GlowTTS+HiFiGAN[17]

Model	**Vocoder**	**Tamil** (Dravidian)			**Hindi** (Indo-Aryan)		
		MCD	F_0	CER	MCD	$\mathbf{F_0}$	CER
FastPitch	HiFiGAN V1	11.19	0.30	0.103	7.59	0.21	0.095
	MB MelGAN	12.00	0.32	0.135	7.79	0.24	0.105
	WaveGrad	16.00	0.30	0.114	9.74	0.20	0.106
Glow-TTS	HiFiGAN V1	11.73	0.33	0.204	8.36	0.24	0.198
	MB MelGAN	12.00	0.32	0.135	8.44	0.27	0.216
	WaveGrad	16.90	0.31	0.243	10.30	0.24	0.191
VITS	-	10.87	0.37	0.295	7.32	0.26	0.176

Table 2. Objective evaluation of a multi-speaker TTS system for different combinations of acoustic models and vocoders for Tamil and Hindi. Here, MB MelGAN refers to Multi-Band MelGAN

3.2. Evaluation of acoustic models and neural vocoders

We evaluate the combinations across acoustic models and vocoders for two languages objectively. In Table 2, we report objective metrics of combining the different acoustic models and vocoders mentioned in Sections 2.1 and 2.2 respectively. Within acoustic models and across languages, we observe a general trend of HiFi-GAN performing better in terms of acoustic metrics with a few exceptions where WaveGrad achieves slightly lower F_0 scores. Further, for each vocoder, FastPitch consistently outperforms Glow-TTS across all three metrics for both the languages. We also observe that VITS achieves the lowest MCD scores, potentially due to it being a fully end-to-end synthesizer. However, in comparison to FastPitch, the VITS model produces less intelligible speech as reflected with larger CER values and with average prosody as given by F_0 scores. Therefore, we pick the combination of FastPitch and HiFiGAN V1 as our model architecture to build TTS systems for Indian languages. The chosen unified architecture is elaborated in Appendix A.1.

[4] https://cloud.google.com/speech-to-text

3.3. Evaluation of training strategies

As discussed earlier, we evaluate the use of supplementary SSIM and ASR loss functions. When using the SSIM loss function, we observed a delayed convergence in the mel-spectrogram reconstruction loss. However, the delay is not significant and both variations converge to similar values. Given no significant advantage, we choose to exclude the additional SSIM loss function. We include the additional ASR loss and obtain intelligibility metrics for the two languages as shown in Table 3. Since there are no significant improvements, we choose to exclude the ASR loss. Thus, we do not add any supplementary loss functions to our training setup.

Language	**Without ASR Loss**	**With ASR Loss**
Tamil (Dravidian)	0.107	0.108
Hindi (Indo-Aryan)	0.094	0.090

Table 3. Objective evaluation on Intelligibility (CER) of our multi-speaker model with and without ASR loss for Tamil and Hindi.

During our experiments, we observed that the alignment loss sharply rises at different points of the training, and subsequently all other losses would rise in response to this. To address this, we experimented with a training schedule where we turned off the aligner loss after 1,500 epochs (roughly after 60%) of the training, and continued to train with other loss functions. We observed that using this training schedule improved the quality for Hindi while not affecting the results for Tamil, and hence we use this for all subsequent models.

3.4. Evaluation of single speaker and multi-speaker models

We train multi-speaker models with one male and one female voice with speaker embeddings jointly learnt and added to the encoder's output. As can be seen in Table 4, multi-speaker models have better scores for both languages and both speakers. This suggests the value of joint training and efficiently deploying models for both male and female speakers. We hypothesize that the improved performance is

because the aligner module not being conditioned on the speaker embeddings learns better alignments on the more diverse data of both genders.

Language	Female		Male	
	Single	Multi	Single	Multi
Tamil (Dravidian)	3.55	3.71	3.84	3.98
Hindi (Indo-Aryan)	3.73	4.02	3.82	3.98

Table 4. Subjective evaluation (MOS) of our single-speaker and multi-speaker models for Tamil and Hindi.

3.5. Evaluation of monolingual and multilingual models

In Table 5, we report the results of the subjective evaluation of multilingual models for the two groups w.r.t. monolingual models. We find it encouraging that multilingual models achieve similar MOS in Kannada, Tamil, Telugu, and Assamese. However, overall the monolingual models outperform in all languages except Gujarati. We thus choose to train monolingual models.

Language	Mono	Multi (Dravidian)	Multi (Indo-Aryan)	Gap
Kannada	3.68	3.60	-	0.08
Malayalam	3.64	3.09	-	0.55
Tamil	3.84	3.80	-	0.04
Telugu	3.66	3.56	-	0.10
Assamese	2.39	-	2.33	0.06
Bengali	3.37	-	2.95	0.42
Gujarati	3.58	-	3.66	-0.08
Hindi	4.00	-	3.19	0.81
Marathi	3.26	-	2.62	0.64
Odia	4.19	-	3.5	0.69

Table 5. Subjective evaluation (MOS) of our monolingual models and multilingual models.

3.6. Comparison of open-source Indic TTS models

Finally, based on our findings, for each of the 13 Indian languages, we train monolingual multi-speaker models with FastPitch and HiFiGAN V1 [5], with no supplementary losses, but with the aligner loss turned off after 1500 epochs. In Table 1, we compare our model against existing open-source TTS models trained on the IndicTTS Dataset. We see that our model is clearly rated better with an average MOS score improvement of 0.51 w.r.t. models proposed in [18] and 0.92 w.r.t. models proposed in [17]. The objective metrics of F_0 and CER also follow a similar trend. The MCD scores however show a lower value for models from [17], but this is uncorrelated to MOS scores [43] and the audio samples are indeed unnatural.

4. CONCLUSION

Neural TTS systems continue to rapidly improve with various changes. We evaluated the choice of acoustic models, vocoders, training strategies, and multi-speaker and multilingual generalizations for Indian languages. With the identified best configuration we train models for 13 Indian languages for both genders and establish that it improves on existing TTS systems. We open-source the models for the 13 languages on the Bhashini platform enabling applications targeting over 1.05 billion native speakers as per 2011 census.

Several directions of future work emerge. Diffusion-based acoustic models and flow-based vocoders need to be compared. Further exploration is required for sharing knowledge while training multilingual TTS models which have a clear advantage in deployability. Open-source models for expressive speech, voice cloning, and unheard speaker generalization for Indian languages remain to be thoroughly investigated.

5. ACKNOWLEDGEMENTS

We would like to thank the Ministry of Electronics and Information Technology (MeitY[6]) of the Government of India and the Centre for Development of Advanced Computing (C-DAC[7]), Pune for generously supporting this work and providing us access to multiple GPU nodes on the Param Siddhi Supercomputer. We would like to thank the EkStep Foundation and Nilekani Philanthropies for their generous grant which went into hiring human resources as well as cloud resources needed for this work. We would like to thank Janki Nawale from AI4Bharat for helping in coordinating the evaluation tasks and extend our gratitude to all the language experts of the AI4Bharat team who helped in the evaluation.

6. REFERENCES

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

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

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

[4] Yi Ren, Yangjun Ruan, Xu Tan, Tao Qin, Sheng Zhao, Zhou Zhao, and Tie-Yan Liu, "Fastspeech: Fast, robust and controllable text to speech," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[5] Adrian Lańcucki, "Fastpitch: Parallel text-to-speech with pitch prediction," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 6588–6592.

[6] Aaron 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," *arXiv preprint arXiv:1609.03499*, 2016.

[7] Yuxuan Wang, RJ Skerry-Ryan, Daisy Stanton, Yonghui Wu, Ron J Weiss, Navdeep Jaitly, Zongheng Yang, Ying Xiao,

[5] https://models.ai4bharat.org/#/tts/samples

[6] https://www.meity.gov.in/

[7] https://www.cdac.in/index.aspx?id=pune

Zhifeng Chen, Samy Bengio, et al., "Tacotron: Towards end-to-end speech synthesis," *arXiv preprint arXiv:1703.10135*, 2017.

[8] Chris Donahue, Julian McAuley, and Miller Puckette, "Adversarial audio synthesis," in *International Conference on Learning Representations*, 2018.

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

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

[11] Geng Yang, Shan Yang, Kai Liu, Peng Fang, Wei Chen, and Lei Xie, "Multi-band melgan: Faster waveform generation for high-quality text-to-speech," in *2021 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2021, pp. 492–498.

[12] Myeonghun Jeong, Hyeongju Kim, Sung Jun Cheon, Byoung Jin Choi, and Nam Soo Kim, "Diff-tts: A denoising diffusion model for text-to-speech," *arXiv preprint arXiv:2104.01409*, 2021.

[13] Vadim Popov, Ivan Vovk, Vladimir Gogoryan, Tasnima Sadekova, and Mikhail Kudinov, "Grad-tts: A diffusion probabilistic model for text-to-speech," in *International Conference on Machine Learning*. PMLR, 2021, pp. 8599–8608.

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

[15] Yuan-Jui Chen, Tao Tu, Cheng-chieh Yeh, and Hung-Yi Lee, "End-to-end text-to-speech for low-resource languages by cross-lingual transfer learning," *Proc. Interspeech 2019*, pp. 2075–2079, 2019.

[16] Edresson Casanova, Julian Weber, Christopher D Shulby, Arnaldo Candido Junior, Eren Gölge, and Moacir A Ponti, "Yourtts: Towards zero-shot multi-speaker tts and zero-shot voice conversion for everyone," in *International Conference on Machine Learning*. PMLR, 2022, pp. 2709–2720.

[17] "Text to speech model training - vakyansh," in *https://github.com/Open-Speech-EkStep/vakyansh-models#tts-models*. Open Speech EkStep.

[18] Anusha Prakash and Hema A Murthy, "Generic indic text-to-speech synthesisers with rapid adaptation in an end-to-end framework," *arXiv preprint arXiv:2006.06971*, 2020.

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

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

[21] Arun Baby, Nishanthi NL, Anju Leela Thomas, and Hema A Murthy, "A unified parser for developing indian language text to speech synthesizers," in *International Conference on Text, Speech, and Dialogue*. Springer, 2016, pp. 514–521.

[22] Nanxin Chen, Yu Zhang, Heiga Zen, Ron J Weiss, Mohammad Norouzi, and William Chan, "Wavegrad: Estimating gradients for waveform generation," *arXiv preprint arXiv:2009.00713*, 2020.

[23] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.

[24] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al., "On the opportunities and risks of foundation models," *arXiv preprint arXiv:2108.07258*, 2021.

[25] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina N. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2018, pp. 4171–4186.

[26] Divyanshu Kakwani, Anoop Kunchukuttan, Satish Golla, Gokul N.C., Avik Bhattacharyya, Mitesh M. Khapra, and Pratyush Kumar, "IndicNLPSuite: Monolingual Corpora, Evaluation Benchmarks and Pre-trained Multilingual Language Models for Indian Languages," in *Findings of EMNLP*, 2020.

[27] Yash Madhani, Sushane Parthan, Priyanka Bedekar, Ruchi Khapra, Vivek Seshadri, Anoop Kunchukuttan, Pratyush Kumar, and Mitesh M Khapra, "Aksharantar: Towards building open transliteration tools for the next billion users," *arXiv preprint arXiv:2205.03018*, 2022.

[28] Marta R Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, et al., "No language left behind: Scaling human-centered machine translation," *arXiv preprint arXiv:2207.04672*, 2022.

[29] Tahir Javed, Sumanth Doddapaneni, Abhigyan Raman, Kaushal Santosh Bhogale, Gowtham Ramesh, Anoop Kunchukuttan, Pratyush Kumar, and Mitesh M Khapra, "Towards building asr systems for the next billion users," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022, vol. 36, pp. 10813–10821.

[30] "Bhashini," in *https://bhashini.gov.in/*. Ministry of Electronics and Information Technology, Government of India.

[31] Masanori Morise, Fumiya Yokomori, and Kenji Ozawa, "World: a vocoder-based high-quality speech synthesis system for real-time applications," *IEICE TRANSACTIONS on Information and Systems*, vol. 99, no. 7, pp. 1877–1884, 2016.

[32] Rohan Badlani, Adrian Lańcucki, Kevin J Shih, Rafael Valle, Wei Ping, and Bryan Catanzaro, "One tts alignment to rule them all," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 6092–6096.

[33] Yi Ren, Xu Tan, Tao Qin, Zhou Zhao, and Tie-Yan Liu, "Revisiting over-smoothness in text to speech," *arXiv preprint arXiv:2202.13066*, 2022.

[34] Yinghao Aaron Li, Ali Zare, and Nima Mesgarani, "Starganv2-vc: A diverse, unsupervised, non-parallel framework for natural-sounding voice conversion," *arXiv preprint arXiv:2107.10394*, 2021.

[35] Andrew Gibiansky, Sercan Ömer Arik, Gregory Frederick Diamos, John Miller, Kainan Peng, Wei Ping, Jonathan Raiman, and Yanqi Zhou, "Deep voice 2: Multi-speaker neural text-to-speech.," in *NIPS*, 2017.

[36] Wei Ping, Kainan Peng, Andrew Gibiansky, Sercan O Arik, Ajay Kannan, Sharan Narang, Jonathan Raiman, and John Miller, "Deep voice 3: 2000-speaker neural text-to-speech," *Proc. ICLR*, pp. 214–217, 2018.

[37] Li Wan, Quan Wang, Alan Papir, and Ignacio Lopez Moreno, "Generalized end-to-end loss for speaker verification," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4879–4883.

[38] Arun Baby, Anju Leela Thomas, NL Nishanthi, TTS Consortium, et al., "Resources for indian languages," in *Proceedings of Text, Speech and Dialogue*, 2016.

[39] Yanxin Hu, Yun Liu, Shubo Lv, Mengtao Xing, Shimin Zhang, Yihui Fu, Jian Wu, Bihong Zhang, and Lei Xie, "Dccrn: Deep complex convolution recurrent network for phase-aware speech enhancement," *arXiv preprint arXiv:2008.00264*, 2020.

[40] Maxim Tkachenko, Mikhail Malyuk, Andrey Holmanyuk, and Nikolai Liubimov, "Label Studio: Data labeling software," 2020-2022, Open source software available from https://github.com/heartexlabs/label-studio.

[41] Robert Kubichek, "Mel-cepstral distance measure for objective speech quality assessment," in *Proceedings of IEEE pacific rim conference on communications computers and signal processing*. IEEE, 1993, vol. 1, pp. 125–128.

[42] Stan Salvador and Philip Chan, "Toward accurate dynamic time warping in linear time and space," *Intelligent Data Analysis*, vol. 11, no. 5, pp. 561–580, 2007.

[43] Elizabeth Salesky, Julian Mäder, and Severin Klinger, "Assessing evaluation metrics for speech-to-speech translation," in *IEEE ASRU*.

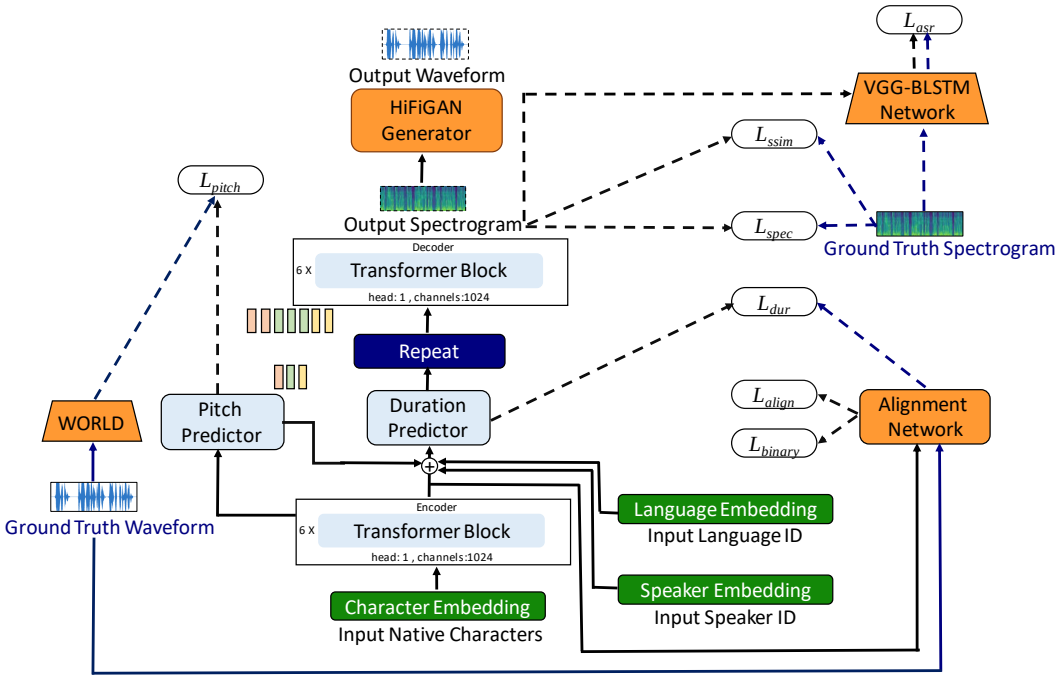

Fig. 2. Unified architecture of our TTS system.

A. APPENDIX

A.1. Unified Architecture

In Figure 2, our final model is depicted, comprising of FastPitch [5] as the acoustic model, the alignment learning framework [32] to map text features to mel-spectrograms and HiFiGAN-V1 [10] as the vocoder.

The encoder maps the input text into a hidden representation $\mathbf{h}$, which is used by the pitch predictor and duration predictor to infer the average pitch and duration for every input symbol respectively. The encoder consists of 6 Transformer blocks each with one head, hidden channel dimension of 1024, and a dropout of 0.1. The pitch predictor and the duration predictor are trained using mean-squared-error losses, L_{pitch} and L_{dur} respectively. The pitch information is added to $\mathbf{h}$ and the resulting vector is discretely upsampled using the duration information. The decoder then operates on the upsampled vector and outputs the generated mel-spectrogram $\hat{\mathbf{m}}$ which is reconstructed, using ground-truth mel-spectrograms $\mathbf{m}$ as a reference, by the reconstruction loss L_{spec} given by,

$$L_{spec} = \|\mathbf{m} - \hat{\mathbf{m}}\|_2^2$$

Unlike the original FastPitch [5], which relies on an external aligner such as a pretrained Tacotron2, we employ an alignment learning framework [32] to align encoded text features $\mathbf{h}$ to mel-spectrograms $\mathbf{m}$. To do this, a soft alignment distribution $\mathcal{A}_{soft}$ based on the learned pairwise affinity between encoded hidden representation and encoded mel-spectrogram frames is first computed and it is normalized using softmax across the domain of input text. An objective function is then used that maximizes the likelihood of the encoded hidden representation given mel-spectrograms using the forward-sum algorithm and its negative is defined as the aligner loss $L_{aligner}$. We also use the Viterbi algorithm as in [32], to convert soft alignments to hard alignments $\mathcal{A}_{hard}$ and use an additional binarization loss L_{binary} that minimizes the KL-Divergence between $\mathcal{A}_{soft}$ and $\mathcal{A}_{hard}$. The aligner is not conditioned on speaker embeddings or language embeddings and takes text embeddings from the text encoder.

$$L_{binary} = -\mathcal{A}_{hard} \odot log\mathcal{A}_{soft}$$

Let L_{ssim} refer to the perceptual SSIM loss briefly discussed in Section 2.3. The ASR loss L_{asr} minimizes the manhattan distance between convolutional features F_{asr} of the ground truth mel-spectrograms $\mathbf{m}$ and synthesized mel-spectrograms $\hat{\mathbf{m}}$, extracted from the intermediate layer before the LSTM layers of a pretrained joint CTC-attention VGG-BLSTM network provided in the Espnet toolkit.

$$L_{asr} = \|F_{asr}(\mathbf{m}) - F_{asr}(\hat{\mathbf{m}})\|_1$$

The final loss $L_{acoustic}$ is given by,

$$L_{acoustic} = L_{spec} + \lambda_{align}L_{align} + \lambda_{dur}L_{dur} + \lambda_{pitch}L_{pitch} + \lambda_{binary}L_{binary} + \lambda_{ssim}L_{ssim} + \lambda_{asr}L_{asr}$$

We set, $\lambda_{dur} = 0.1$, $\lambda_{pitch} = 0.1$, $\lambda_{binary} = 0.1$. For experiments in Section 3.3 we set $\lambda_{ssim} = 1$ and $\lambda_{asr} = 0.5$.

We train our acoustic model and vocoder separately and put them in sequence for inference. We use HiFi-GAN V1 [10] as a vocoder to generate high-fidelity speech from mel-spectrograms. For HiFi-GAN V1 training, we use the ground truth spectrogram as input and learn the generator to produce the ground truth waveform. We train HiFi-GAN with the same objective functions for the generator and discriminators as the original work [10], where the generator objective function is the combination of GAN loss, feature matching loss and the mel-spectrogram loss. Although the vocoder is not conditioned on either speaker or language embeddings, it is able to generalize well.