Curriculum Learning Based Approach for Faster Convergence of TTS Model

Navneet Kaur(✉) and Prasanta Kumar Ghosh

SPIRE Lab, Indian Institute of Science, Bengaluru 560012, India
{knavneet,prasantg}@iisc.ac.in
https://iisc.ac.in/

Abstract. With the advent of deep learning, Text-to-Speech technology has been revolutionized, and current state-of-the-art models are capable of synthesizing almost human-like speech. Recent Text-to-Speech models use a sequence-to-sequence architecture that directly converts text or phoneme sequence into low-level acoustic representation such as spectrogram. These end-to-end models need a large dataset for training, and with conventional learning methodology, they need days of training to generate intelligible and natural voice. 'How to use a large dataset to efficiently train a TTS model?' has not been studied in the past. 'Curriculum learning' has been proven to speed up the convergence of models in other machine learning areas. For TTS task, the challenge in creating curriculum is to establish the difficulty criteria for the training samples. In this paper, we have experimented with various scoring functions based on text and acoustic features and achieved faster convergence of the end-to-end TTS model. We found 'text-length' or the number of phonemes/characters in text to be a simple yet most effective measure of difficulty for designing curriculum for Text-to-Speech task. Using text-length based curriculum, we validated the faster convergence of TTS model using three datasets of different languages.

Keywords: Speech synthesis · Text-to-speech · Curriculum learning · Tacotron

1 Introduction

Text-to-Speech (TTS) is the technology of automatic conversion of text into speech waveform. TTS system aims to resemble, as closely as possible, a native speaker of the language reading that text. A large number of techniques exist in the literature for TTS [2,9,14,23], but the recent advancements in deep learning has revolutionized the field. Today, end-to-end speech synthesis models such as Tacotron [19], TransformerTTS [12], Fastspeech [18] are able to generate human-like voices. These typically include sequence-to-sequence models that convert sequence of characters/phonemes into linear or mel-spetrograms. The spectrograms are then used to generate audio waveforms using Griffin-Lim algortithm

A. Karpov et al. (Eds.): SPECOM 2023, LNAI 14339, pp. 208–221, 2023.
https://doi.org/10.1007/978-3-031-48312-7_17

or neural vocoders such as Wavenet [15], WaveGlow [16], MelGAN [10]. These text-to-spectrograms converters and audio generator models are what comprise an end-to-end TTS system.

However, training end-to-end TTS networks requires a sizable set of studio-quality (text, audio) pairs. Training on huge corpus is slow and it takes days of training to get intelligible and natural speech out of these systems. In this paper, we try to answer: 'How to train an end-to-end TTS model using a large dataset such that it converges faster?' To this end, we exploit curriculum learning techniques.

1.1 Curriculum Learning and TTS

Introduced by Yoshua Benjio in [1], curriculum learning (CL) broadly involves presenting the model with easy examples first and then gradually increasing the level of difficulty of examples. This training strategy has been shown both theoretically and empirically to accelerate the learning of deep learning models in [1,7,22]. [21] provides an extensive survey of CL techniques applied in the fields of computer vision, language processing, and speech recognition. Curriculum learning has demonstrated its effectiveness in improving the generalization capability and convergence rate of models from different domains. In the speech domain, curriculum learning has been used for better generalization, but its use for improving the convergence rate has not been explored. Specifically, curriculum learning has been used for robust far-field speech recognition [17], speech emotion recognition from crowd-sourced labels [13], and pre-training for end-to-end speech translation [20].

At the time of this writing, there is only one paper [8] where CL has been used to develop document-level neural TTS. In this paper, the input samples, i.e. (text, audio) pairs are randomly combined to generate progressively longer sentences in successive epochs of training. This curriculum has helped the model to generalize better and generate speech of duration higher than that available in the training set. The aim of the author in [8] has been to generalize the TTS model to the document level, whereas, in our work, we have made an attempt to use curriculum learning to speed up the convergence of the TTS model. To generalize the TTS to larger text, 'text-length' becomes a natural curriculum criterion, and accordingly, the author in [8] has supplemented the dataset with large text lengths by joining (text, audio) samples. However, in order to speed up the convergence of a TTS model, what criteria would be most effective for sorting the given dataset? In this work, we have experimented with different curriculum criteria and compared their effectiveness in speeding up the convergence of the TTS model. The curriculum criteria we have used are inspired by their success in other domains and we made the first attempt to use these criteria for the TTS learning task. Using the best curriculum criterion, we validated faster convergence of TTS model with three datasets of different languages.

2 Model and Datasets

Due to its popularity and simple yet powerful architecture, we have used Tacotron-2 [19] as Text-to-Spectrogram model for our experiments. Tacotron-2 has LSTM based encoder-decoder architecture with location sensitive attention [3]. It is autoregressive in nature and converts the sequence of characters into mel-scale spectrogram, frame by frame. Specifically, we used Nvidia's Pytorch implementation of Tacotron2 as our TTS model and Griffin-Lim algorithm to vocode the resulting spectrograms.

2.1 Three Datasets

We have used the following datasets to carry out experiments and consolidate our findings.

English Dataset. For English, we have used LJ-Speech dataset. It is a publicly available and most widely used dataset for training end-to-end TTS models. The speaker is an American female who reads passages from non-fiction books. We used 12500 utterances having a total duration of about 24 h as the training set for our experiments. The audio ranged from 1 to 10 s in duration.

Hindi Dataset. To consolidate our findings, we did a few experiments with our lab's Hindi dataset of 12 h duration. The dataset consists of 11,156 audio clips of a single female speaker. Audios are recorded at 16 kHz frequency and vary in length from 1 to 7 s. For text, news data from various publications was used along with school textbooks. The recording was done in 2019 with support from *Gnani.ai* team.

Telugu Dataset. This final dataset used for our experiments is created by our lab as a part of SYSPIN: SYnthesizing SPeech in INdian languages project: *syspin.iisc.ac.in*. We took 10,820 utterances as training data for our experiments which resulted in a total of 38 h of data. The text collected spanned across multiple domains: finance, agriculture, politics, education, health and general. The audios were uttered by a male native speaker of Telugu and the recording was done with help of *Bhashini.ai* team in 2021.

2.2 Metrics for Evaluation

Speech synthesis models are generally evaluated using MOS score. For faster turn-around time, we have extensively used objective measures to evaluate the performance of models. We used the following measures: **Mel Cepstral Distortion** (MCD), **Gross Pitch Error** (GPE), **F0 Frame Error rate** (FFE) [6] and **AlignmenT Score** (ATS). The synthesized mel-spectrograms and audios

were compared with ground truth or recorded ones using MCD, GPE, and FFE, and the monotonicity of generated alignment map was measured using ATS. MCD computes the difference between the ground truth and generated spectrogram in cepstral domain. We use 13 MFCC coefficients and excluded 0th coefficient for MCD calculation as shown in Eq. 1 where $C_{t,k}/\hat{C_{t,k}}$ is k^{th} MFCC coefficient of reference/synthesized spectrogram at frame index 't', where $1 \leq t \leq T$ with T being the total number of frames.

$$MCD = \frac{10\sqrt{2}}{\ln 10}\frac{1}{T}\sum_{t=0}^{T-1}\sqrt{\sum_{k=1}^{13}(C_{t,k} - \hat{C_{t,k}})^2} \tag{1}$$

GPE and FFE are pitch-based measures as defined in Eq. 2 and 3. v_t/ $\hat{v_t}$ are voicing decisions and $p_t/\hat{p_t}$ are pitch values at frame index t in reference/synthesized audio. GPE measures the percentage of voiced frames that deviate by more than 20 percent in the pitch signal of the generated audio compared to the reference audio.

$$GPE = \frac{\sum_t \mathbb{1}[|p_t - \hat{p_t}| > 0.2p_t]\mathbb{1}v_t\mathbb{1}\hat{v_t}}{\sum_t \mathbb{1}v_t\mathbb{1}\hat{v_t}} \tag{2}$$

FFE measures the percentage of frames that either have a 20 percent pitch error or a differing voicing decision between the synthesized and reference audio. F0 contours for audio are obtained using PRAAT software. Since synthesized and ground truth sequences could be different in length, we used dynamic time warping with *l2* distance as the distance measure to time align both mel-spectrograms and pitch contours before comparing them for MCD, GPE and FFE computation.

$$FFE = \frac{1}{T}\sum_t(\mathbb{1}[|p_t - \hat{p_t}| > 0.2p_t] + \mathbb{1}[v_t \neq \hat{v_t}]) \tag{3}$$

For tracking the convergence of the model, we also use the AlignmenT Score (ATS) of the generated spectrograms. This is defined as the normalized sum of attention weights that lie in the diagonal region of the alignment matrix as shown in Fig. 1. Here, slope Tq/Tv is used to find the diagonal, and the sum of weights in the region 'c' distance away from the diagonal is calculated. We used c=5 frames for our calculations and computed alignment score as shown in Eq. 4 where Tq is sum of all attention weights or equivalently generated spectrogram length. ATS measures the sharpness and monotonicity of the attention maps. A higher ATS score indicates model has learned the alignment well.

$$ATS = \frac{\sum Attention\_weights\_within\_diagonal\_region}{Tq} \tag{4}$$

For final comparison, we also conducted **Mean Opinion Score** (MOS) test, remotely through Google Form. The evaluators were presented with a few sentences and corresponding audios generated by models trained using different

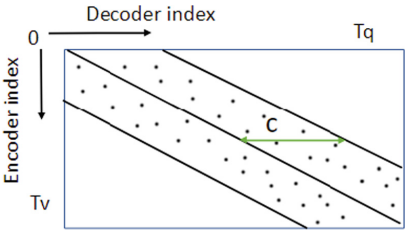

Fig. 1. Computation of Alignment Score: Tq is the number of spectrogram frames, and Tv is character length.

curricula. Different curriculum audios corresponding to the same text were kept together, to bring out relative comparison. To avoid any bias, the order of different curriculum audios was randomly shuffled for each sentence. Also, since some audios may sound very similar, we did not want to burden listeners with hard ranking. Instead, they were asked to rate each audio on a scale of 1 to 5 in terms of naturalness to capture preferences at a finer level.

3 Curriculum Learning Criteria

To apply curriculum learning to any task we need to address two critical questions: how to rank the training examples, and how to modify the sampling procedure based on this ranking. Thus, depending on the application, we need to define two functions: i) Scoring function [1], and ii) Pacing function [1]. To speed up the learning for the TTS task, we experimented with the following scoring functions to rank (text, audio) training examples.

3.1 Text-Length

In neural machine translation tasks, 'text-length' is shown to be an effective measure of the difficulty of training samples. Since, an end-to-end TTS model involves text encoding as that in a neural machine translator, we believed that this intuitive measure of difficulty may be helpful for Text-to-Speech task as well. Specifically, we computed text-length as number of characters in text input of training sample. Being a text-based feature, 'text-length' can be computed for the dataset even before the audio is recorded and thus, can be beneficial in TTS deployment as discussed in Sect. 5.

3.2 Acoustic Feature

We also experimented with an acoustic feature and explored its use for speeding up the convergence of a TTS model. Work done in data selection for TTS in [4,5,11] suggests that utterances with low articulation and low F0 standard deviation generate better-sounding samples when used to train a TTS model.

Here, articulation is defined as shown in Eq. 5 where total energy is computed as the sum of squares of audio samples, and average speaking rate is calculated as the number of vowels divided by utterance length. We experimentally verified the results of the data-selection study using our datasets and found that even an end-to-end TTS model (Tacotron-2) generates more natural voice when trained on a data subset with lower articulation and lower F0 standard deviation. This alludes to the fact that the model learns better and faster if the training samples have low values of articulation and F0 standard deviation. Thus, we constructed an 'acoustic feature' to select audios with low values of these features, as defined in Eq. 6, and used this 'acoustic feature' as a measure of difficulty for implementing CL for TTS.

$$articulation = \frac{total\_energy}{average\_speaking\_rate} \tag{5}$$

$$acoustic\_feature = articulation * F0\_standard\_deviation \tag{6}$$

3.3 Automatic Curriculum Learning

Conventional or pre-defined CL requires us to define the scoring function to rank order the training samples. But, what may be easy for humans may not be easy for the model. Thus, automatic curriculum learning was introduced in which the ranking of samples is model-driven and not human knowledge-driven. As automatic CL is proved to be more advantageous over pre-defined CL in literature, we also experimented with this strategy. Specifically, we trained our TTS model over the entire training corpus for 100 epochs and used this partially trained model to generate mel-spectrograms for all training samples. We then computed DTW-aligned mean squared error distance between synthesized and ground truth mel-spectrograms. Using this distance, we rank-ordered the training examples. The lower the distance, the easier the training sample.

4 Experiments and Results

We primarily used English dataset, LJ-Speech for obtaining the most appropriate difficulty measure for TTS task. We then verified the results on two other data sets: Hindi and Telugu.

4.1 Result on LJ-Speech Dataset

We began the experimentation with LJ-Speech dataset. For each scoring function discussed in section-3, we implemented double step pacing function as follows:
i) We use easiest 8 h of data (as per the scoring function) and trained model for 10k iterations,
ii) Use easiest 15 h of data and trained model for 20k more iterations,
iii) Finally, entire 24 h corpus is used to train the model for further 30k iterations.

As a baseline, we implemented random curriculum in which data subsets are chosen randomly. Table 1 shows the performance of models after each step of curriculum, i.e., after 10k, 30k and 60k iterations. For reference, we have also included results of model trained on 'full data'(entire 24 h corpus) at various stages without any curriculum. We found that for acoustic-feature based curriculum, GPE score which measures the naturalness of speech has reduced the most in 10k iterations, but MCD score remains poor till 30k iterations. On the other hand, we note that after both 10k and 30k iteration points, the MCD score corresponding to 'Text-length' curriculum is the lowest. Also, the GPE score which is poor for 'Text-length' after 10k, is significantly improved or reduced after 30k iterations. 'Text-length' seems to be more effective criterion than the acoustic feature for faster convergence. ATS results favor automatic 'DTW-MSE' based curriculum. The learning curves on validation data are shown in Fig. 2. Considering both the objective scores and learning curves, we find that curriculum learning indeed benefits the convergence and model trains faster as compared to the random curriculum. Especially, 'Text-length' and 'DTW-MSE' based CL prove to be more beneficial. Although DTW-MSE gave the best results, it is an automatic CL and we need to train the model on the complete dataset for implementing this curriculum. On the other hand, 'Text-length' is an easily computed pre-defined feature whose performance is competent with that of a dynamic DTW-MSE feature-based curriculum. We found 'Text-length' to be the most effective and efficient curriculum criteria for accelerated convergence. We thus validate the efficacy of 'Text-length' based curriculum on two other datasets.

Table 1. Performance of models trained using different curricula after 10k/30k/60k iterations. Bold entries correspond to best scoring curriculum.

Feature	MCD	GPE	FFE	ATS
Full-data(No CL)	50.75/32.77 /32.60	0.317/0.175 /0.198	0.211/0.121 /0.137	0.012/0.055 /0.075
Random	54.12/43.12 /32.29	0.343/0.228 /0.194	0.221/0.156 /0.141	0.009/0.010 /0.105
Text-length	**49.57/31.90** /31.75	0.326/0.202 /0.196	0.231/0.145 /0.134	0.016/0.018 /0.037
Acoustic-feature	51.96/36.19 /32.85	**0.269**/0.250 /0.195	0.182/0.163 /0.132	0.014/0.032 /0.030
DTW-MSE	50.75/33.54 /31.67	0.290/**0.186** /0.197	**0.173/0.134** /0.134	**0.017/0.0.056** /0.053

MOS Score Test: To subjectively evaluate the performance of 'text-length' based curriculum versus random curriculum, we conducted MOS score test. For this, we used models after second stage of double step pacing function experiment, i.e. models trained for 30k iterations. For comparison, we also used model trained using entire 23 h data for 30k iterations without curriculum. We synthesized 10 sentences using each of three models, and used 30 synthesized audios for the test. We ensured that the length of test sentences has wide enough range so that 'text-length' curriculum based model trained on shorter sentences gets no undue advantage. Total 46 listeners participated and average scores are shown in Table 2. The results are in agreement with objective evaluation and 'text-length'

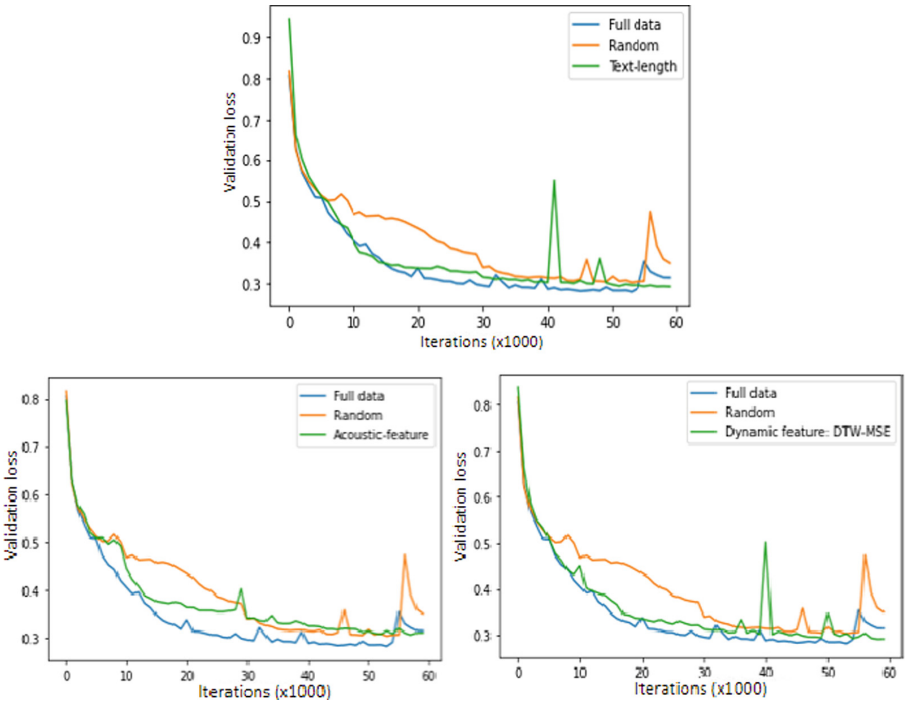

Fig. 2. Validation loss for different curricula: Text-length(top), acoustic feature(bottom-left), DTW-MSE feature(bottom-right) on LJ-Speech.

feature based curriculum has obtained similar MOS score with 15 h data as that of vanilla trained model on 23 h data. The MOS score obtained by random curriculum is, however, very low as the speech synthesized was not very intelligible. The leftmost column in Fig. 3 shows the alignment of a test sentence synthesized by the three models. While 'text-length' based model has achieved sharp monotonic alignment, the alignment obtained from random curriculum trained model still looks blur and hazy.

Table 2. MOS score results for different models trained for 30k iterations. 'Max. duration' is total duration of data the model has seen till this stage of training.

Curriculum	Max. duration	MOS score
Full-data(No CL)	23 h	3.5455
Random	15 h	1.8286
Text-length	15 h	3.5956

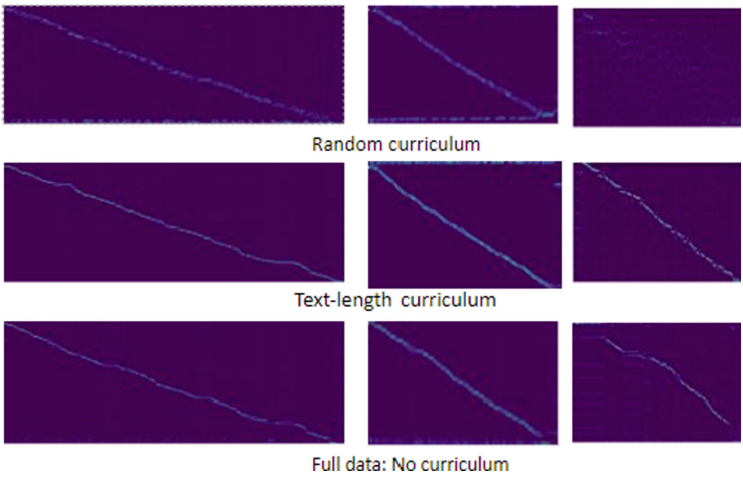

Fig. 3. Attention matrices generated by models trained using different curricula: LJ-Speech (leftmost), Hindi (middle) and Telugu(rightmost).

4.2 Results on Hindi Dataset

We consolidated our results using a Hindi dataset with a total duration of 11.7 h. We implemented 'text-length' based curriculum as follows:
i) We trained Tacotron-2 model for 20k iterations on 5 h easiest data,
ii) Then, followed by 20k iterations on 8 h of easiest data.

Figure 4 shows the validation loss curve and Table 3 shows the objective evaluation results. We observed that the maximum gain of using curriculum learning is observed in the initial phase of training as 'text length' curriculum-based model shows the best performance in terms of all the objective metrics after 15k iterations.

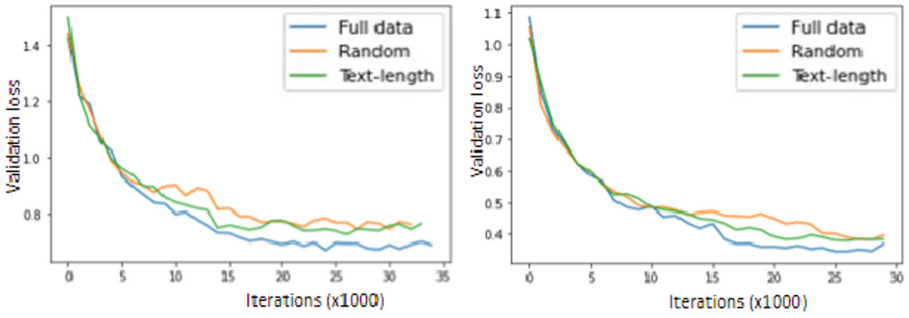

Fig. 4. Validation loss for Hindi(left) and Telugu(right) dataset.

Table 3. Performance of different curriculum models on Hindi dataset. In score format 'x/y//z', 'x/y' represent the score after training model for 15k/20k iterations on 5 h subset, 'z' is the score after further training for 20k iterations with 8 h subsets.

Curriculum	MCD	GPE	FFE	ATS
Full-data	45.37/34.31 //32.27	0.133/0.088 //0.080	0.133/0.060 //0.063	0.036/0.083 //0.109
Random	48.57/35.41 //32.00	0.125/0.088 //0.063	0.117/0.084 //0.059	0.018/0.017 //0.017
Text-length	**41.68**/34.49 //32.46	**0.113**/0.091 //0.063	**0.097**/0.091 //0.060	**0.050**/0.053 //0.085

MOS Score Test: We also conducted MOS score test for Hindi dataset. Ten sentences of varied lengths were synthesized by models trained for 20k iterations on a) 5 h random subset, b) 5 h of shortest text-length subset, and c) full 11.7 h of data. Total 79 responses were recorded and the average scores are reported in Table 4. The middle column in Fig. 3 shows the alignment for a test sentence synthesized by three models. The observations are in agreement with ATS & MOS scores and it is visually clear that the attention of text length curriculum-based model is sharper and better than that obtained by random curriculum model.

Table 4. MOS score results for models trained for 20k iterations on different curricula for Hindi dataset.

Curriculum	Max. duration	MOS score
Full-data(No CL)	11.7 h	3.8202
Random	5 h	1.9759
Text-length	5 h	3.2240

4.3 Results on Telugu Dataset

The total duration of Telugu dataset was 38 h. To experiment with text-length based curriculum learning, we trained Tacotron-2 model for 20k iterations on 10 h easiest data, followed by 10k iterations on 15 h of easier data. Figure 4 shows that the validation loss of text-length based curriculum train remains lower than random curriculum. Also, Table 5 shows that 'Text-length' based curriculum achieves best objective scores, MCD & ATS after 10k/20k iterations, again highlighting that model learns faster using this curriculum.

MOS Score Test: For Telugu dataset, we conducted MOS score test for two stages of learning. MOS-1 and MOS-2 present the scores obtained after 1st and 2nd phase of training. Seven sentences for each stage were synthesized and presented to listeners. A total of 45 responses were collected and results are shown in Table 6. 'Full-data' scores are reported after vanilla training the model for 20k

and 30k iterations using the complete 38 h of data. MOS-1 is higher for vanilla learning as compared to curriculum learning as data observed by curriculum learning model is just about 10 h, as compared to 38 h in vanilla training model. MOS-2 however, indicates that using just 39% of training data, i.e. 15 h data, 'text-length' CL based model achieved MOS score competent with that of vanilla training model.

Table 5. Performance of different curriculum models on Telugu dataset; Here, in score format x/y//z, x/y is the score after training for 10k/20k iterations on 10 h subset; z is the score after further training for 10k iterations with 15 h subsets.

Curriculum	MCD	GPE	FFE	ATS
Full-data	40.59/**32.13** //31.25	**0.155/0.143** //0.126	0.104/0.082 //0.087	6.13e-5/0.012 //0.022
Random	37.83/36.71 //30.85	0.158/0.153 //0.133	0.093/0.091 //0.084	7.04e-5/4.04e-5 //9.48e-4
Text-length	**34.40**/33.52 //28.92	0.173/0.151 //0.134	0.104/0.084 //0.079	**1.11e-4/3.93e-4** //1.98e-3

Table 6. Telugu dataset: MOS-1: model trained for 20k iterations on 10 h subset; MOS score-2: model further trained on 15 h data for 10k iterations.

Curriculum	MOS-1 (Max. data)	MOS-2 (Max. data)
Full-data(No CL)	3.3932 (38 h)	3.6015 (38 h)
Random	2.0673 (10 h)	3.2118 (15 h)
Text-length	2.7138 (10 h)	3.5096 (15 h)

5 Discussion

By experimenting with multiple datasets, we found that, we are able to achieve faster convergence using 'text-length' based curriculum as compared to random curriculum. At the same time, 'text-length' based curriculum learning achieves a similar MOS score as that of conventional learning using a significantly lesser amount of data.

We want to highlight that 'text-length' based curriculum learning can provide practical advantages while deploying a TTS system as follows. Conventional TTS system involves text collection, and weeks of audio recording before it is used to train a TTS model. Being a text-based measure, 'text length' can be computed before the audio is recorded. Thus, we can begin the recording with shorter/easier text, and use it to train the model. We can progressively increase the length of the text to be recorded until the model has generalized and the MOS score requirement is achieved. This enables us to achieve faster convergence and record just the sufficient amount of data required to train a TTS model, thus, reducing the cost of data creation.

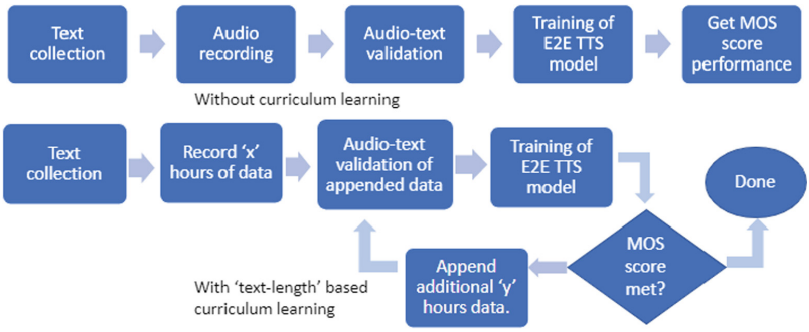

Fig. 5. Conventional TTS system deployment(top) and curriculum enabled TTS deployment(bottom).

As shown in Fig. 5, a conventional TTS system involves text collection, audio recording, and the laborious task of audio-text validation before it is used to train a TTS model. While deploying TTS using a 'text-length' based curriculum, we can begin recording a small chunk of x hours of data, and then keep adding y hours of data until the MOS requirement is achieved. Curriculum learning gives us a criterion to record data in chunks so that we collect just the sufficient amount of data needed to train the model. This also relieves the burden of manual validation of unduly large amounts of data.

6 Conclusion and Future Scope

In this paper, we have established that 'text-length' is an appropriate difficulty measure for curriculum learning in TTS task. We have demonstrated that 'text-length' based curriculum learning helps speed up the convergence of a sequence-to-sequence based Text-to-Speech model on three datasets.

We have worked with Tacotron-2 model which is most widely used text-to-spectrogram model provided by Google, but the results can be extended to other TTS models as well. Work can be done to check the effectiveness of the proposed methods for other auto-regressive and non auto-regressive models for spectrogram generation. Neural vocoder is crucial component of end-to-end TTS system. Even though it does not need paired (text, audio) data for training; it still needs a large amount of audio data and weeks of training to generate high-quality voice. We can explore the use of acoustic feature-based or utterance duration based curriculum learning for vocoders to speed up their training. In this work, we have restricted ourselves to the fixed training schedule. As 'DTW-MSE' loss based curriculum gave positive results on LJSpeech data set, we can work further in the direction of automatic curriculum learning for TTS. We can optimize the training schedule or update it dynamically with the help of model feedback. Finally, we have used simple and clean TTS datasets in our work. In the future, the power of curriculum learning can be explored using other complex datasets.

References

1. Bengio, Y., Louradour, J., Collobert, R., Weston, J.: Curriculum learning. In: Proceedings of the 26th Annual International Conference on Machine Learning, pp. 41–48 (2009)
2. Black, A., Tokuda, K., Zen, H.: An hmm-based speech synthesis system applied to English. In: Proceedings of (2002)
3. Chorowski, J.K., Bahdanau, D., Serdyuk, D., Cho, K., Bengio, Y.: Attention-based models for speech recognition In: Advances in Neural Information Processing Systems 28 (2015)
4. Cooper, E., Levitan, Y., Hirschberg, J.: Data selection for naturalness in hmm-based speech synthesis. In: Speech Prosody (2016)
5. Cooper, E., Wang, X.: Utterance selection for optimizing intelligibility of TTS voices trained on ASR data. Interspeech **2017**, 1 (2017)
6. Fang, W., Chung, Y.A., Glass, J.: Towards transfer learning for end-to-end speech synthesis from deep pre-trained language models. arXiv preprint arXiv:1906.07307 (2019)
7. Hacohen, G., Weinshall, D.: On the power of curriculum learning in training deep networks. In: International Conference on Machine Learning, pp. 2535–2544. PMLR (2019)
8. Hwang, S.W., Chang, J.H.: Document-level neural TTS using curriculum learning and attention masking. IEEE Access **9**, 8954–8960 (2021)
9. Jurafsky, D., Martin, J.H.: Speech and language processing: an introduction to natural language processing, computational linguistics, and speech recognition
10. Kumar, K., et al.: Melgan: generative adversarial networks for conditional waveform synthesis. In: dvances in Neural Information Processing Systems 32 (2019)
11. Kuo, F.Y., Ouyang, I.C., Aryal, S., Lanchantin, P.: Selection and training schemes for improving TTS voice built on found data. In: INTERSPEECH, pp. 1516–1520 (2019)
12. Li, N., Liu, S., Liu, Y., Zhao, S., Liu, M.: Neural speech synthesis with transformer network. In: AAAI Conference on Artificial Intelligence, vol. 33, pp. 6706–6713 (2019)
13. Lotfian, R., Busso, C.: Curriculum learning for speech emotion recognition from crowdsourced labels. IEEE/ACM Trans. Audio, Speech, Lang. Process. **27**(4), 815–826 (2019)
14. Onaolapo, J., Idachaba, F., Badejo, J., Odu, T., Adu, O.: A simplified overview of text-to-speech synthesis. Proc. World Congr. Eng **1**, 5–7 (2014)
15. Oord, A.v.d., et al.: Wavenet: a generative model for raw audio. arXiv preprint arXiv:1609.03499 (2016)
16. Prenger, R., Valle, R., Catanzaro, B.: Waveglow: a flow-based generative network for speech synthesis. In: International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 3617–3621. IEEE (2019)
17. Ranjan, S., Hansen, J.H.: Curriculum learning based approaches for robust end-to-end far-field speech recognition. Speech Commun. **132**, 123–131 (2021)
18. Ren, Y., Hu, C., Tan, X., Qin, T., Zhao, S., Zhao, Z., Liu, T.Y.: Fastspeech 2: fast and high-quality end-to-end text to speech. arXiv preprint arXiv:2006.04558 (2020)
19. Shen, J., et al.: Natural TTS synthesis by conditioning wavenet on mel spectrogram predictions. In: International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 4779–4783. IEEE (2018)

20. Wang, C., Wu, Y., Liu, S., Zhou, M., Yang, Z.: Curriculum pre-training for end-to-end speech translation. arXiv preprint arXiv:2004.10093 (2020)
21. Wang, X., Chen, Y., Zhu, W.: A survey on curriculum learning. IEEE Trans. Pattern Anal. Mach. Intell. **44**(9), 4555–4576 (2021)
22. Weinshall, D., Cohen, G., Amir, D.: Curriculum learning by transfer learning: Theory and experiments with deep networks. In: International Conference on Machine Learning, pp. 5238–5246. PMLR (2018)
23. Wu, Z., Watts, O., King, S.: Merlin: an open source neural network speech synthesis system. In: SSW, pp. 202–207 (2016)