DURATION MODELING OF NEURAL TTS FOR AUTOMATIC DUBBING

Johanes Effendi, Yogesh Virkar, Roberto Barra-Chicote, Marcello Federico*

Amazon AI

{yvvirkar|rchicote|marcfede}@amazon.com

ABSTRACT

Automatic dubbing (AD) addresses the problem of translating speech in a video with speech in another language while preserving the viewer experience. A most important requirement of AD is isochrony, i.e. dubbed speech has to closely match the timing of speech and pauses of the original audio. In our automatic dubbing system, isochrony is modeled by controlling the verbosity of machine translation; inserting pauses in the translations, a.k.a. prosodic alignment; and controlling the duration of text-to-speech (TTS) utterances. The latter two steps heavily rely on speech duration information, either to predict or control TTS duration. So far, duration prediction was based on a proxy method while duration control on linear warping of the TTS speech spectrogram. In this study, we propose novel duration models for neural TTS that can be leveraged both to predict and control TTS duration. Experimental results show that compared to previous work, the new models improve or match the performance of prosodic alignment and significantly enhance neural TTS speech quality for both slow and fast speaking rates.

Index Terms— speech translation, text-to-speech, automatic dubbing, duration modelling

1. INTRODUCTION

Automatic dubbing (AD) aims to automatically translate and replace the speech contained in a video with speech in a different language, while making the result sound and look as natural as the original. Hence, especially when the speaker is on screen, dubbed speech should match not only the meaning of the original speech but also its prosodic structure. In particular, a crucial requirement of AD is isochrony [2], which is when dubbed speech perfectly matches the timing of the original utterances and pauses. This factor makes AD different from speech translation, because it adds a temporal aspect into the translation and speech generation processes.

In order to achieve isochrony, our AD pipeline generates translations that closely match the length of the source transcript [3]. Then, each translation is segmented into phrases that reflect the temporal prosodic structure – i.e. speech and pauses– of the original audio. This step is called prosodic alignment (PA). After that, target language speech is generated using a neural TTS model by taking into account timing constraints for each phrase and pause. In other words, the TTS speaking rate of each phrase is adjusted in order to make the speech segment exactly fit the timing of the corresponding source phrase. When deciding the segmentation, PA trades off isochrony with speech fluency by optimizing the position of the split points and by possibly introducing a small relaxation to the time boundaries when needed [1].

*The work was done as an intern in Amazon.

The PA and TTS components heavily rely on duration information both of the source sentence and the translation. PA makes use of TTS duration and speaking rate information estimates while searching for the optimal segmentation. On the other hand, TTS uses duration information to generate speech segments with a desired timing. However, in our current pipeline [1], PA and TTS use two independent approaches. In PA, duration predictions of TTS for each segmentation hypothesis are obtained from a pre-computed duration of each word inside the target sentence. These timings are obtained after running TTS on the entire sequence without any pause, force aligning the audio with text, and computing the duration of each word from the time stamps. This solution is clearly a proxy of the real measure, as it does not take into account small changes in phoneme duration when pauses are inserted before or after words. On the other hand, running TTS over each segmentation hypothesis explored during PA would clearly be very inefficient. On the side of TTS, duration control of a generated utterance is based on linearly warping the Mel spectrogram generated by the TTS Tacotron model when generating speech at a normal speaking rate.

In this paper, we present novel duration models of TTS speech that are both efficient and precise when used to perform predictions, and very detailed when used to drive TTS generation. In particular, we upgraded our neural Tacotron model by providing explicit duration control at the phoneme-level. The underlying assumption of this work is that both PA and TTS would benefit from sharing a single duration model (DM) both for predicting and controlling neural TTS duration. In the following, we describe our dubbing architecture, overview related work about duration modeling for TTS, describe our proposed duration models and how they are integrated into our AD architecture, and provided experimental results on AD of TED talks in two directions: English-Spanish and English-Italian.

2. DUBBING ARCHITECTURE

We build on the automatic dubbing architecture presented in [4, 5]. Figure 1 shows (in green) our previous work to extend a speech-to-speech translation [6, 7, 8] pipeline with neural machine translation (MT) robust against ASR errors and able to control verbosity of the output [9, 10, 11]; prosodic alignment (PA) [1, 4, 12] which addresses phrase-level synchronization of the MT output by leveraging the force-aligned source transcript; neural text-to-speech (TTS) [13, 14, 15] with precise duration control; and, finally, audio rendering that enriches TTS output with the original background noise (extracted via audio source separation with deep U-Nets [16, 17]) and reverberation, estimated from the original audio [18, 19]. We show (in red) how we additionally add the duration model component to support PA in predicting phrase durations and to support TTS to perform explicit duration control.

To appear in *Proc. ICASSP 2022, May 22-27, 2022, Singapore.*

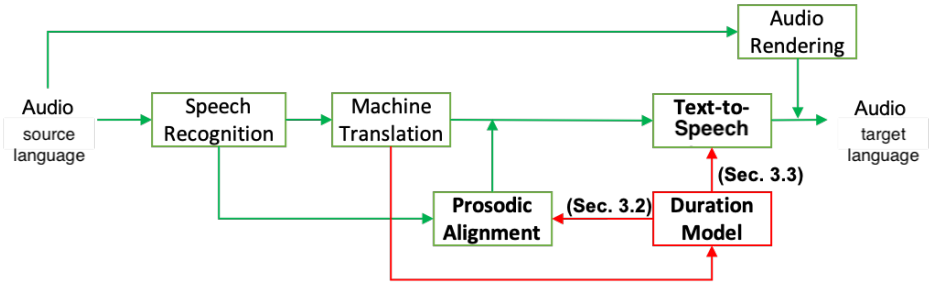

Fig. 1. Virkar et al. (2021) automatic dubbing pipeline [1] (green) with the proposed decoupled duration model (red) and upgraded components (in bold).

3. RELATED WORK

In the early days of generative modelling for TTS, duration information was modelled intrinsically in the hidden Markov model [20], while there was also an attempt to separate it into an explicit model at the state level [21, 22]. Other research contributions explored the impact of linguistic descriptors as an input of Artificial Neural Networks to predict the phone durations [23, 24] for guiding downstream tasks, for example, guiding concatenative TTS systems [25] or modelling phone durations in emotional speech [26]. Explicit duration modelling is also found in the recent neural-based TTS systems. Compared with the attention-based Tacotron model [27, 28] which models the duration implicitly through attention, explicit duration modelling is used in FastSpeech [29, 30] and Parallel Tacotron [31, 32]. They reported that duration information contributes to faster inference due to phoneme state multiplication and also improved naturalness. In this work, we show that duration information can also be used for duration control to improve speech quality, a concept which to the best of our knowledge has not yet been explored. In fact, our approach is relevant in any time-constrained TTS scenarios, such as AD, where the duration of the target utterance is predefined. In our study, in particular, we investigate the use of such TTS models in the AD task by leveraging a duration model to maintain isochrony.

4. DURATION MODEL

Duration Models (DM) learn to predict the duration d_n of each phoneme x_n inside a sequence of phonemes $\boldsymbol{x} = [x_0, \ldots, x_n, \ldots, x_{N-1}]$ representing a text to be synthesized with a neural TTS model. We approach this task with a sequence-to-sequence model, which maps the sequence $\boldsymbol{x}$ into a sequence $\boldsymbol{d} = [d_0, \ldots, d_n \ldots, d_{N-1}]$, where each d_n represents duration information corresponding to x_n. We employ an architecture similar to that of the encoder of the Tacotron 2 model [28], which has been proven to effectively encode sequence of phonemes. Our architecture is illustrated in Figure 2. After applying an embedding layer on the phoneme sequence $\boldsymbol{x}$, the resulting hidden representation is encoded through a series of 1-dimensional convolutional layers, a bidirectional long-short term memory (LSTM) layer, and finally a linear output layer. In the following, we describe two versions of DM learning to predict different types of duration information through specific loss functions.

4.1. DM predicting number of frames (frmDM)

The output layer of this DM version has a scalar output, which is the duration of each phoneme in the input sequence expressed in number of frames. This DM employs the L2 loss in the logarithmic domain so that:

$$L_{\boldsymbol{d}} = \sum_{n=0}^{N-1} \|\log(\hat{d}_n), \log(d_n)\|^2, \quad (1)$$

where $\hat{d}_n$ is the predicted duration and d_n is the target duration. To normalize the duration of an utterance with predicted durations $\hat{\boldsymbol{d}} = [\hat{d}_0, \hat{d}_1, \ldots, \hat{d}_{N-1}]$ to a total target duration D^T, each duration entry is scaled by the coefficient ρ calculated as follows:

$$\rho = \frac{D^T}{\sum_{n=0}^{N-1} \hat{d}_n}, \quad (2)$$

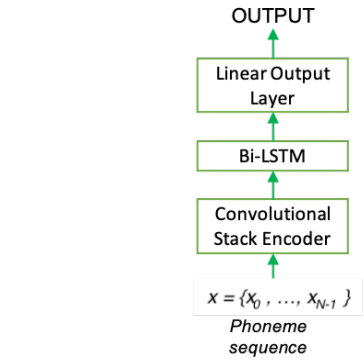

Fig. 2. Our proposed duration model.

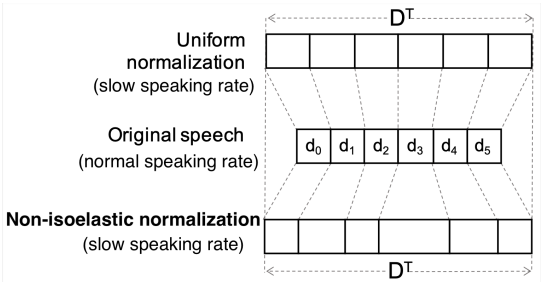

Fig. 3. Difference between uniform normalization and non-isoelastic normalization to increase duration. Both utterances have the same target duration D^T, but with different treatments on the phoneme-level duration.

According to this method, each phoneme duration is normalized uniformly. We call this method **uniform normalization** (see Fig. 3).

4.2. DM predicting duration distribution (gauDM)

Phoneme state or phoneme duration are often modelled as a Gaussian distribution [33, 34]. For this version of DM, duration of a phoneme is defined as:

$$d_n = \mu_n + \rho \cdot \sigma_n \tag{3}$$

For this version, we use two output layers which, respectively, output two scalars $\hat{\mu}$ and $\hat{\sigma}$ for each phoneme, resulting in the output sequence $\hat{\boldsymbol{d}} = [(\hat{\mu}_0, \hat{\sigma_0}), (\hat{\mu}_1, \hat{\sigma_1}), \ldots, (\hat{\mu}_{N-1}, \hat{\sigma_{N-1}})]$. We define the loss function as the Gaussian negative log-likelihood following eq. (4):

$$\begin{aligned} L_{\boldsymbol{d}} &= -\log \hat{\boldsymbol{p}}(\boldsymbol{d}|\boldsymbol{x}) = -\sum_{n=0}^{N-1} \log \hat{p}(d_n) \\ &= \frac{N}{2}\log 2\pi + \frac{1}{2}\sum_{n=0}^{N-1}\left(\log \hat{\sigma}_n^2 + \frac{(d_n - \hat{\mu}_n)^2}{\hat{\sigma}_n^2}\right) \end{aligned} \tag{4}$$

This model enables **non-isoelastic normalization** (see Fig. 3), where each phoneme duration is adjusted independently, based on the variance σ_n and the coefficient ρ, which can be calculated from the target duration D^T as follows:

$$\rho = \left(D^T - \sum_{n=0}^{N-1} \hat{\mu}_n\right) \Big/ \sum_{n=0}^{N-1} \hat{\sigma}_n \tag{5}$$

Notice that according to this formulation, a normal speaking rate corresponds to a coefficient $\rho = 0$, slower speech corresponds to $\rho < 0$, and faster speech to $\rho > 0$.

5. DURATION MODEL INTEGRATION

We review here how the prosodic alignment and TTS components take advantage of the duration models, respectively, to predict and control TTS duration.

5.1. Duration Model for Prosodic Aligner

We consider here the prosodic alignment model in [1] which applies dynamic programming to segment a target sequence of words on the basis of a log-linear combination of five kinds of features:

1. Speaking rate variation across consecutive target segments
2. Speaking rate match across corresponding source and target segments
3. Isochrony score for left and right relaxations
4. Language model score of target break point
5. Cross-lingual semantic match score across corresponding source and target segments

From these features, 1-3 rely heavily on duration information of target TTS. We replaced the duration predictions based on precomputed values (see above) with predictions computed on the fly by our DM on each segmentation hypothesis generated by the PA model.

5.2. Duration Model for Explicit Duration TTS

We use our off-the-shelf implementation of explicit duration TTS, with a structure similar to Parallel Tacotron [31]. The duration modelling works by multiplying the encoded phoneme state based on the predicted duration, before passing it to the decoder. In their recent upgrade to the model [32], the authors showed a possibility to explicitly control the speech duration to be slower or faster. Unfortunately, they scaled all predicted durations by a fixed factor, which caused all durations to be scaled uniformly (see Eq. 2). In this study, we incorporate the duration from the non-isoelastic gauDM model to get better naturalness in slower and faster speech.

6. EVALUATION DATA AND METRICS

For PA, we use the same re-translated and annotated dataset from [1] of 495 video clips extracted from 20 TED Talks available for English-Italian (en-it) and English-Spanish (en-es). Each video clip contains a single sentence having at least one pause of at least 300 ms. Pauses are detected on the source side by force-aligning original English audio with text [35]. For the duration model and the TTS model, we use our internal dataset which consists of 13,237 and 3,304 conversational Italian and Spanish utterances with a total of 19.1 and 7.5 hours respectively.

We use the full-fledged PA model from [1] that, besides the aforementioned features, also includes a time-boundary relaxation mechanism. We train PA models in a similar manner using cross-validation with stratified sampling. In [1], the PA model predicted phrase-level durations from precomputed word-level durations. Word-level timings were obtained by running TTS on the entire utterance without any pause and force-aligning audio with text. We term this approach as TTS + FA (force alignment). In this work, we train PA models by predicting durations using trained duration models, namely FrmDM and GauDM. For both DM models, we use a hidden size of 512 units and bidirectional LSTMs. Both the DM and TTS model are trained with Adam optimizer with learning rate 1e-3.

To estimate quality of segmentation output by PA trained using different duration prediction methods, we use Accuracy, Smoothness and Fluency metrics defined in [4] and additionally define the following metric:

Speaking rate deviation measures the absolute difference of speaking rate (SR) from normal SR (SrD_{norm}) or from the source SR (SrD_{src}). Lower values for both versions of this metric are preferred since lower deviation of SR from normal SR can improve speech fluency, and a lower deviation from source SR implies improved speaking rate match score for PA [1].

Since the PA model relies on either TTS + FA, FrmDM or GauDm for estimating duration and hence SR, it is important to use the same method for computing automatic metrics related to SR so that the comparisons are fair. For this purpose, we use TTS+FA method to estimate SR when computing metric values.

7. EXPERIMENTS

7.1. Automatic Evaluation

Table 1 compares performance of PA models trained using different methods for duration prediction including the previous method TTS + FA [1] and the proposed approaches FrmDM (Sec. 4.1) and GauDM (Sec. 4.2). We use the automatic metrics described in Sec. 6. For both en-it and en-es, all three PA models have a similar accuracy

		TTS + FA [1] Previous	FrmDM Proposed	GauDM Proposed
en-it	Accuracy ↑	80.8	**81.4**	81.0
	Smoothness ↑	**72.5**	71.4	70.9^{a}
	Fluency ↑	**63.4**	62.6	61.4
	SrD_{norm} ↓	25.9	**25.5**	25.8
	SrD_{src} ↓	25.0	$\mathbf{23.3}^{a}$	24.0^{a}
en-es	Accuracy ↑	**81.6**	81.2	79.6
	Smoothness ↑	75.4	73.9	$\mathbf{76.8}^{ab}$
	Fluency ↑	48.3	49.7	$\mathbf{53.5}^{ab}$
	SrD_{norm}	37.3	36.4	$\mathbf{34.8}^{ab}$
	SrD_{src} ↓	40.5	38.9	$\mathbf{37.6}^{ab}$

Table 1. Automatic evaluation for Prosodic alignment trained with different methods for duration prediction - (a) TTS + FA [1], (b) FrmDM (Sec. 4.1) and (c) GauDM (Sec. 4.2). Significance testing is done at levels $^{a} = p < 0.01$ against TTS+FA; $^{b} = p < 0.01$ against FrmDM.

		Slow Speech		Fast Speech	
	Normalization	Wins	Score	Wins	Score
en-it	Uniform	21.2	4.55	14.7	6.22
	Non-isoelastic	34.2^{*}	4.70	40.1^{*}	6.69
en-es	Uniform	29.4	5.05	24.6	5.89
	Non-isoelastic	37.9^{*}	5.19	44.9^{*}	6.25

Table 2. Subjective evaluation result comparing speech synthesized with uniform or non-isoelastic normalization (gauDM) for both slow speech and fast speech conditions. Significance testing is done at levels $^{*} = p < 0.01$.

score. For en-it, we observe that neither FrmDM nor GauDM provide any improvements in Fluency and Smoothness. On the other hand, both FrmDM and GauDM provide relative improvements for SrD_{norm} (+1.5%, +0.4%) and SrD_{src} (+6.8%, +4%) respectively. For en-es, we observe that FrmDM and GauDM obtain relative improvements in Fluency (+2.9%, +10.8%), SrD_{norm} (+2.4%, +6.7%), SrD_{src} (+4%, +7.2%) respectively with improvements for SrD_{src} being statistically significant. Additionally, GauDM improves Smoothness by +1.9%. All improvements for GauDm measured against both the baseline and FrmDM are statistically significant [36] at $p < 0.01$. Additionally we note that both proposed duration models significantly improve the time cost for PA model training by roughly 100x. This shows that our proposed approaches have the potential to improve SR related features while not degrading accuracy and making PA model training significantly faster.

7.2. Human Evaluation

To further test the efficacy of the non-isoelastic DM we perform a human evaluation selecting 50 sentences at random from the test set. We synthesize an audio corresponding to each test sentence using two different conditions. In the *uniform* condition, for each sentence we perform TTS using normal SR and obtain corresponding audio. This is followed by a morphing step that uniformly normalizes phoneme durations to achieve a desired SR. In the *non-isoelastic* condition, for each input sentence GauDm computes phoneme duration distributions which are then normalized differently (see Eq. (3)) based on the desired SR. This duration information is then input to a TTS model to produce the final audio.

For the dubbing use case, human evaluations done in the past [1] show that the dubbing quality is impacted by the TTS component. One issue in this regard was the *uniform* stretching of phonemes to achieve desired SR. We hypothesize that the non-isoelastic stretching solution can provide improved naturalness. To test this, we conduct two separate audio evaluations by testing the two conditions for slow and fast speech. For each sentence, we fix SR value for both conditions and draw SR at random from the sets $\{0.6, 0.7, 0.8, 0.9\}$ and $\{1.1, 1.2, 1.3, 1.4\}$ for testing slow and fast speech respectively.

For each language, we ask native speakers to grade the listening experience for audio from each condition using a 0-10 scale. We run evaluations using Amazon Mechanical Turk with 20 subjects grading all 50 audios for a total of 1,000 ratings. We compare the *uniform* and *non-isoelastic* conditions using the Wins (percentage of times one condition is preferred over the other) and Score (average score of dubbed videos) metrics.

Table 2 shows the results. For both languages, we observe that the non-isoelastic condition is preferred for both slow and fast speech. In particular we obtain improvements in Wins for slow speech (en-it: +61.3%, en-es: +28.9%) and fast speech (en-it: +172.8%, en-es: +82.5%) with all results being statistically significant ($p < 0.01$). Additionally, we found that this improvement holds for all ranges of speaking rates with higher improvements at the extremes. We also obtain relative improvements in subjective score for slow speech (en-it: +3.3%, en-es: +4.3%) and fast speech (en-it: +2.8%, en-es: +7.6%). Thus our proposed approach shows great potential to be applicable for the dubbing use case in which target phrases often need to be synthesized with slower or faster than normal SR owing to the difference in verbosity between the source and target languages.

8. CONCLUSIONS

To summarize, in our automatic dubbing pipeline both the prosodic alignment and the TTS components require speech duration information to either predict or control TTS duration. In previous work, we utilized two separate TTS models for providing such duration information. Additionally, for PA we used a proxy method that involves first running TTS on the entire text sequence without any pause, force aligning audio with text and computing duration of each word from the time stamps generated by force alignment. Compared to this proxy method for predicting duration, our proposed duration models help to significantly speed up PA model training and inference (100x) while mostly improving or matching the performance for both Italian and Spanish. To study the duration control of neural TTS, we conduct human evaluations comparing uniform normalization against the proposed non-isoelastic normalization. Results show that audio rendered with non-isoelastic normalization is preferred for both slow and fast speech with statistical significance for both languages, thus demonstrating the utility of proposed duration models for automatic dubbing.

9. ACKNOWLEDGEMENT

We thank Goeric Huybrecths for providing resources for TTS model training.

10. REFERENCES

[1] Y. Virkar, M. Federico, R. Enyedi, and R. Barra-Chicote, "Improvements to Prosodic Alignment for Automatic Dubbing," in *Proc. ICASSP*, 2021, pp. 7543–7574.

[2] F. Chaume, "Synchronization in dubbing: A translation approach," pp. 35–52, 2004.

[3] S. M. Lakew, M. Federico, Y. Wang, C. Hoang, Y. Virkar, R. Barra-Chicote, and R. Enyedi, "Machine Translation Verbosity Control for Automatic Dubbing," in *Proc. ICASSP*, 2021, pp. 7538–7542.

[4] M. Federico, Y. Virkar, R. Enyedi, and R. Barra-Chicote, "Evaluating and optimizing prosodic alignment for automatic dubbing," in *Proc. Interspeech*, 2020, p. 5.

[5] M. Federico, R. Enyedi, R. Barra-Chicote, R. Giri, U. Isik, A. Krishnaswamy, and H. Sawaf, "From Speech-to-Speech Translation to Automatic Dubbing," in *Proc. IWSLT*, 2020, pp. 257–264.

[6] F. Casacuberta, M. Federico, H. Ney, and E. Vidal, "Recent efforts in spoken language translation," *IEEE Signal Processing Magazine*, vol. 25, no. 3, pp. 80–88, May 2008.

[7] R. J. Weiss, J. Chorowski, N. Jaitly, Y. Wu, and Z. Chen, "Sequence-to-Sequence Models Can Directly Translate Foreign Speech," in *Proc Interspeech*, 2017, pp. 2625–2629.

[8] L. Vila, C. Escolano, J. Fonollosa, and M. Costa-Jussà, "End-to-End Speech Translation with the Transformer," in *Proc. IberSPEECH*, 2018, pp. 60–63.

[9] S. M. Lakew, M. Di Gangi, and M. Federico, "Controlling the Output Length of Neural Machine Translation," in *Proc. of IWSLT*, 2019.

[10] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention Is All You Need," *arXiv:1706.03762 [cs]*, 2017.

[11] M. Di Gangi, R. Enyedi, A. Brusadin, and M. Federico, "Robust Neural Machine Translation for Clean and Noisy Speech Transcripts," in *Proc. IWSLT*, 2019.

[12] A. Öktem, M. Farrús, and A. Bonafonte, "Prosodic Phrase Alignment for Machine Dubbing," in *Proc. of Interspeech*, Graz, Austria, 2019.

[13] N. Prateek, M. Lajszczak, R. Barra-Chicote, T. Drugman, J. Lorenzo-Trueba, T. Merritt, S. Ronanki, and T. Wood, "In Other News: a Bi-style Text-to-speech Model for Synthesizing Newscaster Voice with Limited Data," in *Proc. NAACL*, 2019, pp. 205–213.

[14] J. Latorre, J. Lachowicz, J. Lorenzo-Trueba, T. Merritt, T. Drugman, S. Ronanki, and K. Viacheslav, "Effect of data reduction on sequence-to-sequence neural TTS," *arXiv:1811.06315 [cs, eess]*, Nov. 2018, arXiv: 1811.06315.

[15] J. Lorenzo-Trueba, T. Drugman, J. Latorre, T. Merritt, B. Putrycz, R. Barra-Chicote, A. Moinet, and V. Aggarwal, "Towards Achieving Robust Universal Neural Vocoding," in *Proc. Interspeech*, 2019.

[16] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Proc. ICMAI*, 2015, pp. 234–241.

[17] A. Jansson, E. Humphrey, N. Montecchio, R. Bittner, A. Kumar, and T. Weyde, "Singing Voice Separation with Deep U-NET Convolutional Networks," in *Proc. of the 18th International Society for Music Information Retrieval Conference*, 2017.

[18] Heiner Löllmann, Emre Yilmaz, Marco Jeub, and Peter Vary, "An improved algorithm for blind reverberation time estimation," in *Proc. IWAENC*, 2010, pp. 1–4.

[19] Emanuel AP Habets, "Room impulse response generator," Tech. Rep. 2.4, Technische Universiteit Eindhoven, 2006.

[20] H. Zen, K. Tokuda, and A. W. Black, "Statistical parametric speech synthesis," *Speech Commun.*, vol. 51, pp. 1039–1064, 2009.

[21] J. V. Santen, "Assignment of segmental duration in text-to-speech synthesis," *Comput. Speech Lang.*, vol. 8, pp. 95–128, 1994.

[22] Shun-Zheng Yu, "Hidden semi-markov models," *Artificial Intelligence*, vol. 174, no. 2, pp. 215–243, 2010.

[23] R. Córdoba, J. M. Montero, J. M. Gutiérrez, J. A. Vallejo, E. Enriquez, and J. M. Pardo, "Selection of the most significant parameters for duration modelling in a spanish text-to-speech system using neural networks," *Computer Speech Language*, vol. 16, no. 2, pp. 183–203, 2002.

[24] G. E. Henter, S. Ronanki, O. Watts, M. Wester, Z. Wu, and S. King, "Robust TTS duration modelling using dnns," in *Proc. ICASSP*, 2016, pp. 5130–5134.

[25] M. Mohri, C. Allauzen, and M. Riley, "Statistical modeling for unit selection in speech synthesis," in *Proc. ACL*, 2004, pp. 55–62.

[26] A. Lazaridis, T. Ganchev, T. Kostoulas, I. Mporas, and N. Fakotakis, "Phone duration modeling: Overview of techniques and performance optimization via feature selection in the context of emotional speech," *Int. J. Speech Technol.*, vol. 13, pp. 175–188, 2010.

[27] Y. Wang et. al., "Tacotron: Towards end-to-end speech synthesis," in *Proc Interspeech*, 2017.

[28] J. Shen, R. Pang, R. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerry-Ryan, R. Saurous, Y. Agiomyrgiannakis, and Y. Wu, "Natural TTS synthesis by conditioning wavenet on mel spectrogram predictions," in *Proc. ICASSP*, 2018.

[29] Yi Ren, Yangjun Ruan, Xu Tan, Tao Qin, Sheng Zhao, Zhou Zhao, and Tie-Yan Liu, "Fastspeech: Fast, robust and controllable text to speech," in *Proc. NEURIPS*, 2019.

[30] Y. Ren, C. Hu, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T. Liu, "Fastspeech 2: Fast and high-quality end-to-end text to speech," *arXiv preprint arXiv:2006.04558*, 2020.

[31] I. Elias, H. Zen, J. Shen, Y. Zhang, Y. Jia, R. Weiss, and Y. Wu, "Parallel tacotron: Non-autoregressive and controllable tts," in *Proc. ICASSP*, 2021, pp. 5709–5713.

[32] I. Elias, H. Zen, J. Shen, Y. Zhang, Y. Jia, R. Skerry-Ryan, and Y. Wu, "Parallel tacotron 2: A non-autoregressive neural tts model with differentiable duration modeling," *arXiv preprint arXiv:2103.14574*, 2021.

[33] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Duration Modeling For HMM-Based Speech Synthesis," in *ICSLP*, 1998.

[34] D. Moungsri, T. Koriyama, and T. Kobayashi, "Duration prediction using multiple gaussian process experts for gpr-based speech synthesis," in *Proc. ICASSP*, 2017, pp. 5495–5499.

[35] R. M. Ochshorn and M. Hawkins, "Gentle Forced Aligner," 2017.

[36] E.W. Noreen, *Computer-Intensive Methods for Testing Hypotheses: An Introduction*, Wiley, 1989.