INTERSPEECH 2023
20-24 August 2023, Dublin, Ireland

Explicit Intensity Control for Accented Text-to-speech

Rui Liu[1], *Haolin Zuo*[1], *De Hu*[1], *Guanglai Gao*[1] *and Haizhou Li*[2,3]

[1]Inner Mongolia University, China
[2]Shenzhen Research Institute of Big Data, School of Data Science,
The Chinese University of Hong Kong, Shenzhen, China
[3] National University of Singapore, Singapore

liurui_imu@163.com, zuohaolin_0613@163.com, cshood@imu.edu.cn, csggl@imu.edu.cn,
haizhouli@cuhk.edu.cn

Abstract

Accented text-to-speech (TTS) synthesis seeks to generate speech with an accent (L2) as a variant of the standard version (L1). How to control the intensity of accent is a very interesting research direction. Recent works design a speaker-adversarial loss to disentangle the speaker and accent information, and then adjust the loss weight to control the accent intensity. However, there is no direct correlation between the disentanglement factor and natural accent intensity. To this end, this paper proposes a new intuitive and explicit accent intensity control scheme for accented TTS. Specifically, we first extract the posterior probability from the L1 speech recognition model to quantify the phoneme accent intensity for accented speech, then design a FastSpeech2 based TTS model, named Ai-TTS, to take the accent intensity expression into account during speech generation. Experiments show that our method outperforms the baseline model in terms of accent rendering and intensity control.

Index Terms: Accented, Text-to-Speech (TTS), Intensity, Explicit Control

1. Introduction

Accented text-to-speech (TTS) synthesis aims to synthesize speech with foreign accent instead of native speech [1]. Note that accent is characterized by a distinctive manner of expression that is influenced by the mother tongue, social group of speakers, or spoken in a particular region [2]. Therefore, the wide adoption of speech applications, such as chatbot and movie dubbing, calls for the study of accented TTS synthesis [3]. Another important practical application is in the design of tools aimed at improving L2 phonological acquisition in language learners.

For accented TTS, some attempts tried to model the accent expression through model interpolation [4–6], variance information prediction [4–6], specific quinphone linguistic features [7, 8] and tone/stress embedding [8, 9], etc. However, the accent as perceived in human speech is subtle and at a fine level [10]. How to control the intensity of an accent is still an open challenge [11]. Wutiwiwatchai et al. [12] proposed an accent level adjustment mechanism for bilingual TTS synthesis, where the accent level is adjusted by means of interpolation between HMMs of native phones and HMMs of corresponding foreign phones. This method provides an effective fine-grained accent intensity control scheme, while it cannot be used in current deep learning TTS models [13–16], such as Tacotron [9, 17] and FastSpeech [18, 19] based architectures. In a recent deep learning based multilingual TTS study [1], the authors employed the domain adversarial training [20] to disentangle the accent identity from the speaker identity where the accent level can be controlled by varying the domain adversarial weight [1]. Such an adversarial weight method controls the utterance-level accent intensity of speech by using the model hyper-parameter. There is no direct and measurable correlation between the controlling factor and the natural accent intensity. The question is how to characterize the fine-grained phoneme-level accent intensity meaningfully, and employ the intensity to control the synthesis of L2 speech for state-of-the-art TTS models, which is the focus of this paper.

Fortunately, we found that there is a great deal of work in the field of Computer-aided pronunciation training (CAPT) [21, 22] to measure the pronunciation of non-native learner. Most of these works assumed that acoustic properties in the learner's pronunciation are similar to a native English speaker's acoustics when their pronunciation similarity is high and vice-versa. Considering this, for each phoneme's in a learner's utterance, a representative score based on posterior probability of the phoneme models given uttered phoneme speech acoustics, called as *Goodness of Pronunciation* (GoP) [23, 24] was proposed and achieved remarkable performance.

Inspired by this, we propose a FastSpeech2 based accented TTS model, named *Ai-TTS*, which synthesizes L2 speech by conditioning phoneme-wise accent intensity information. To quantify the fine-grained accent intensity, we utilize a pretrained L1 speech recognition model to calculate the GoP score as the phoneme intensity score for each L2 phoneme. During inference, we can control accent expression easily by conditioning intensity labels manually. The experimental results show that our system successfully achieves better accent expressiveness and controllability than the baseline system.

The significant contributions of this work include, 1) We introduce a novel FastSpeech2 based accented TTS synthesis paradigm, named Ai-TTS, that explicitly controls the accent intensity in output speech; 2) We successfully design and implement a fine-grained accent intensity quantization method with accent speech recognition model; 3) We show that the proposed Ai-TTS framework outperforms the baseline models and generates high-quality L2-accented speech.

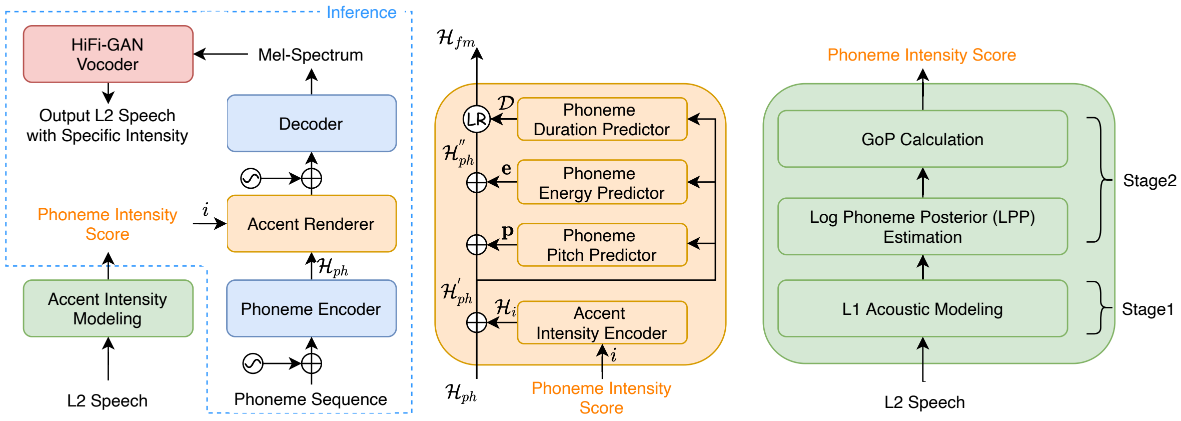
2. Ai-TTS: Methodology

2.1. Model Architecture

We propose a neural architecture, termed as Ai-TTS, as shown in Fig.1 (a) that consists of an accent intensity modeling

This research is funded by the High-level Talents Introduction Project of Inner Mongolia University (No. 10000-22311201), the National Natural Science Foundation of China (NSFC) (No. 62206136, 62271432), and Agency for Science, Technology and Research (A*STAR) under its AME Programmatic Funding Scheme (Project No. A18A2b0046).

10.21437/Interspeech.2023-2270



(a) Ai-TTS Architecture (b) Accent Renderer (c) Accent Intensity Modeling

Figure 1: *The diagrams for the proposed Ai-TTS system. (a) shows the overall model architecture; (b) illustrates the detailed structure of the accent renderer, which includes the accent intensity encoder and the variance adaptor; (c) demonstrates the workflow of the accent intensity modeling.*

module, a phoneme encoder, an accent renderer, and a decoder. The phoneme encoder and decoder are implemented on the basis of FastSpeech2 [18]. The novel accent intensity modeling module aims to learn the phoneme-level accent intensity score for the input L2 speech. Note that the accent intensity modeling module can be treated as the preprocessing operation to label the phoneme intensity score for L2 speech dataset. The phoneme encoder encodes the input phoneme sequence into phoneme embedding. The accent renderer seeks to modulate the input phoneme embeddings with the learned phoneme intensity score and various variance information (including pitch, energy and duration) towards the target accent. Note that the input phoneme intensity score enables all variance information to be affected by the fine-grained accent intensity. The decoder converts the modulated phoneme embeddings into a mel-spectrum sequence. Finally, the universal HiFi-GAN vocoder [25] is used to synthesize high-quality L2 speech.

2.2. Accent Renderer

The traditional variance adaptor in [18] just adds different variance information such as duration, pitch and energy into the phoneme embeddings, that lacks an accent controlling mechanism. We note that our accent renderer augments the phoneme embeddings with the phoneme accent intensity scalar. The accent renderer provides phoneme-level accent information according to fine-grained accent intensity. As shown in Fig.1 (b), the accent renderer consists of 1) an accent intensity encoder, 2) a phoneme pitch predictor, 3) a phoneme energy predictor, and 4) a phoneme duration predictor.

Assume that phoneme embedding $\mathcal{H}_{ph}$ is the phoneme encoder output and the learned phoneme intensity score is i. We implement the accent intensity encoder with a linear layer to transform a real-valued accent intensity score i to an intensity embedding vector, $\mathcal{H}_i$. Afterwards, the phoneme-wise intensity embedding $\mathcal{H}_i$ is concatenated to the phoneme embedding $\mathcal{H}_{ph}$ to form the accented phoneme embedding $\mathcal{H}^{'}_{ph}$. The phoneme pitch and energy predictors take $\mathcal{H}^{'}_{ph}$ as input and are expected to output more accurate pitch and energy information, that are $\mathbf{p}$ and $\mathbf{e}$ respectively, for L2 speech. We sum the accented phoneme, pitch and energy embeddings to form an augmented accented phoneme embedding $\mathcal{H}^{''}_{ph} = \mathcal{H}^{'}_{ph} + \mathbf{p} + \mathbf{e}$. A length regulator (LR) is used to transform the $\mathcal{H}^{''}_{ph}$ to frame-level embeddings $\mathcal{H}_{fm}$ based on the phoneme duration $\mathcal{D}$ predicted by duration predictor.

In a nutshell, the accent renderer learns to project the desired accent and its intensity into the input phoneme embedding $\mathcal{H}_{ph}$. The phoneme-level real-value score i of accent intensity, ranging from 0 to 1, is generated by a novel accent intensity modeling module, which will be described in Sec. 2.3.

2.3. Accent Intensity Modeling

As mentioned in Sec.1, inspired by the CAPT field, we first pretrain the native speech recognition network with the L1 acoustic model, and then quantify the accent intensity for the phoneme sequence of L2 speech by comparing it with the posterior probability between L2 and L1 phonemes. As shown in Fig.1(c), the accent intensity modeling is conducted in two stages: 1) Stage1: L1 acoustic modeling stage, and 2) Stage2: accent intensity quantization stage.

2.3.1. Stage1: L1 Acoustic Modeling

In this work, we employ the Time-Delay Neural Network (TDNN) based acoustic model [26] , for modelling long term temporal dependencies from short-term acoustic features, with the L1 speech dataset. The TDNN acoustic model consists of 6 TDNN layers and softmax layer. The Mel Frequency Cepstral Coefficients (MFCC) and i-vector [27] features are extracted as the TDNN input, that are acoustic observations sequences $\mathbf{o}$ as well. The initial TDNN layers learn the narrow contexts and the deeper TDNN layers process the hidden activations from a wider temporal context. As the output, the last softmax layer of TDNN acoustic model can directly output the posterior $\mathcal{P}(\cdot)$ of each phoneme of input speech. More details are referred to [28].

After TDNN pretraining, the trained acoustic model takes the phoneme segments of the L2 speech as input to calculate the GoP score, that regarded as the accent intensity score for each phoneme. We will describe it next.

2.3.2. Stage2: Accent Intensity Quantization

To quantify the accent intensity score for all phonemes of L2 speech, the trained TDNN acoustic model takes the L2 speech, instead of L1 speech, as input to extract the posterior for each phoneme p. Afterwards, following [29], the Log Phoneme Posterior (LPP) ratio between the canonical phoneme p and the one phoneme q with the highest score is used to approximate the GoP score:

$$\mathrm{GOP}(p) = \log \frac{\mathrm{LPP}(p)}{\max_{q \in Q} \mathrm{LPP}(q)} \tag{1}$$

$$\mathrm{LPP}(p) = \log \mathcal{P}\left(p \mid \mathbf{o}; t_s, t_e\right) \tag{2}$$

where Q is the whole phoneme set. $\mathbf{o}$ is the input acoustic observations. t_s and t_e are the start and end frame indexes, obtained by forced-alignment, respectively. $\mathcal{P}(p)$ means the prior of phoneme p. Note that the straight way to approximate the LPP(p) of phoneme segment p is by averaging the frame based posterior $\mathcal{P}(s_t|\mathbf{o})$ [29]:

$$\begin{aligned} \mathrm{LPP}(p) &\approx \frac{1}{t_e - t_s + 1} \sum_{t=t_s}^{t_e} \log \mathcal{P}\left(p \mid o_t\right) \\ \mathcal{P}\left(p \mid o_t\right) &= \sum_{s \in p} \mathcal{P}\left(s \mid o_t\right) \end{aligned} \tag{3}$$

where s_t is the senone class label [30] of the frame t generated by force alignment with the given canonical phoneme p. $s|s \in p$ is the states belonging to those triphones whose current phone is p.

At last, we follow [31] and normalize the GoP score to [0,1], with 1 as the strongest intensity, as the final accent intensity score i for accent rendering during TTS.

2.4. Run-time Inference

During inference, the Ai-TTS takes the phoneme sequence and synthesizes the controllable L2 speech by conditioning the custom phoneme intensity score manually to achieve explicit intensity control for accented TTS. When all phonemes share a score, it can be viewed as utterance-level control.

3. Experiments and Results

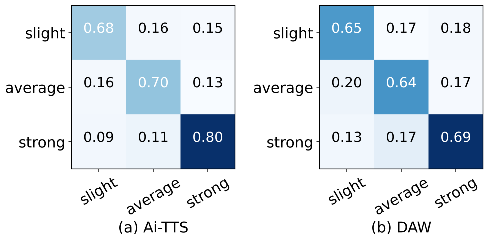
3.1. Datasets

L1 Speech Dataset: LibriSpeech corpus [32] is derived from audiobooks that include reading-style speech recorded by 2238 native L1 English speakers, which contains 960 hours of data in the train set. All audios are sampled at 16 kHz and coded in 16 bits. We adopt the "train_960_cleaned" subset to conduct the TDNN acoustic modeling.

L2 Speech Dataset: We train Ai-TTS on the publicly available L2-ARCTIC corpus [33], which includes about 26 hours recordings of accented English from 24 non-native speakers, whose are native in *Hindi, Korean, Mandarin, Spanish, Arabic and Vietnamese*. Two male and two female speakers contributed in each language. In L2-ARCTIC, scripts and their phoneme-level alignment annotations are provided. The speech data is sampled at 44.10 kHz and coded in 16 bits. For Ai-TTS training, we select the subset of *Mandarin* accent and partition the speech data into training, validation, and test sets at a ratio of 8:1:1.

3.2. Experimental Setup

The phoneme encoder and decoder of Ai-TTS use 6 Feed-Forward Transformer (FFT) blocks. The dimension of the phoneme embedding $\mathcal{H}_{ph}$ is 256. The phoneme sequence is generated by the grapheme to phoneme (G2P) conversion toolkit [1]. The decoder generates an 80-channel mel-spectrum, which is extracted with 12.5ms frame shift and 50ms frame length, as output. We downsampled all speech files to 22.05 kHz and trimmed leading and trailing silence. In accent renderer, the accent intensity encoder consists of a linear layer, which encodes the accent intensity score i into a 256 dimensional $\mathcal{H}_i$.



(a) Ai-TTS (b) DAW

Figure 2: *Confusion matrices between perceived and intended accent intensity categories of synthesized speech. (a) Ai-TTS; b) DAW. The X-axis and Y-axis of the figures represent the perceived and intended categories, namely slight, average, and strong.*

We use the Adam optimizer [34] with β_1 = 0.9, β_2 = 0.98 and follow the same learning rate schedule in [35]. All models are trained with 900k steps to ensure complete convergence. The codes are written in Python 3.6 using the Pytorch library 1.7.0. The GPU type is NVIDIA Tesla P100 with 24GB GPU memory. We employ a pretrained universal HiFI-GAN [25] vocoder for waveform generation.

For TDNN acoustic modeling, we extract the 100 dimensional i-vector and 400 dimensional MFCC features as the TDNN input [2]. The acoustic frame context configuration of TDNN are [-1,0,1], [-1,0,1], [-3,0,3], [-3,0,3], [-3,0,3], [-6,-3,0] in order. We follow the Kaldi script [3] to train TDNN with 128 batch size. The word error rate of the TDNN acoustic model achieved 5.21 % for various test sets of LibriSpeech on average, which is encouraging. The following subsection will investigate the explicit controllability performance for accented TTS.

3.3. Controllability Evaluation on Utterance-level

In this section, we conduct subjective experiments to validate our Ai-TTS by comparing Ai-TTS with the domain adversarial weight (DAW) control mechanism [1]. Different from the phoneme-level intensity control method of Ai-TTS, DAW uses adversarial weights to control the utterance-level accent intensity. However, if we use one value to define all the phoneme intensity scores in an utterance, it can simulate utterance-level intensity control. To this end, to verify the better interpretability of our explicit intensity control method, we compare the utterance-level intensity control effects of Ai-TTS and DAW for fair comparison.

Note that DAW is an utterance-level control method, we set the intensity of all phonemes in Ai-TTS to same value to achieve utterance-level control. We first conduct an accent intensity classification experiment. Specifically, for DAW, we follow [1] and set the adversarial weight from 0 to 0.1. We consider the weight value from 0 to 0.03 as 'slight', 0.04 to 0.06 as 'average' and 0.07 to 0.1 as 'strong' in three categories.

[1] https://github.com/Kyubyong/g2p

[2] https://github.com/kaldi-asr/kaldi/blob/master/egs/librispeech/s5/run.sh

[3] https://github.com/kaldi-asr/kaldi/blob/master/egs/librispeech/s5/local/nnet3/tuning/run_tdnn_1b.sh

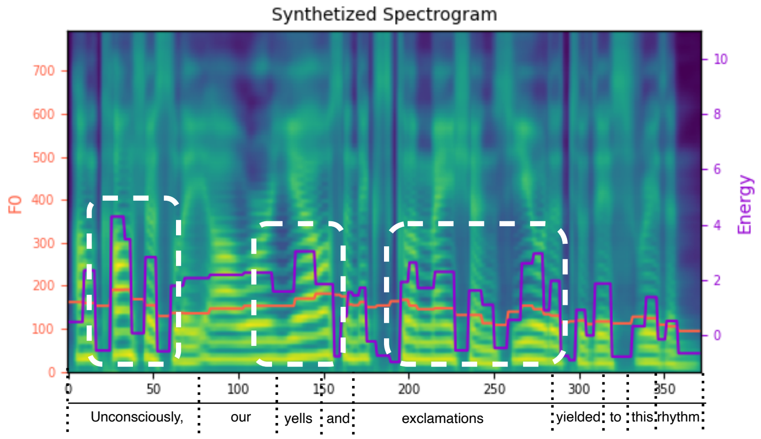


Figure 3: *Spectrogram, F0 contour and energy of an utterance with specific phoneme-level intensity score. While boxes indicates the acoustic performance with higher accent intensity. It is observed that Ai-TTS provides an effective explicit accent intensity control mechanism.*

Table 1: *The comparison of the accent expression for different systems, including expressiveness MOS score, standard deviation (σ), skewness (γ), kurtosis ($\mathcal{K}$) and average dynamic time warping (DTW) distances (ϱ) for pitch.*

System	MOS (↑)	σ (↑)	γ (↓)	$\mathcal{K}$ (↓)	ϱ (↓)
Ground Truth	4.43 ± 0.025	48.5	0.586	0.935	NA
FastSpeech2 [1]	3.86 ± 0.031	41.3	0.632	0.992	19.22
Ai-TTS (i=1) *(proposed)*	**4.01 ± 0.022**	**47.2**	**0.561**	**0.944**	**18.56**

For Ai-TTS, we consider the intensity scores from 0 to 0.3 as 'slight', 0.4 to 0.6 as 'average' and 0.7 to 1 as 'strong'. We select 100 utterances from the test set, resulting in 100 samples for both systems. Accordingly, all listeners are instructed to rate the accent intensity category, that are 'slight', 'average' or 'strong', for each sample. A listener can listen to the samples multiple times when needed.

Fig. 2 reports the intensity confusion matrices. We can find that the Ai-TTS system shows a higher correlation between the perceived and intended accent intensity categories, with a correlation of over 80%, that is considered a competitive result against other intensity-controlled studies [1]. Furthermore, the Ai-TTS system clearly outperforms the contestant. The experiment results confirm the superiority of the proposed explicit accent intensity control mechanism.

3.4. Controllability Evaluation on Phoneme-level

We further evaluate the intensity-controlled speech at phoneme level. Note that F0 (or pitch) and energy are related to accent expression [1]. Fig. 3 shows an example of spectrogram, F0 contour and energy of an utterance "Unconsciously, our yells and exclamations yielded to this rhythm.". Due to space limitations, the phoneme sequence "AH2 N K AA1 N SH AH0 S L IY0 sp AW1 ER0 Y EH1 L Z AE1 N D sp EH2 K S K L AH0 M EY1 SH AH0 N Z sp Y IY1 L D IH0 D T UW1 DH IH1 S R IH1 DH AH0 M" is omitted. We assign the intensity score of phonemes of words "Unconsciously", "yells" and "exclamations" as 0.9 while those of other phonemes are 0.1. The white boxes in Fig. 3 show that the phonemes with higher accent intensity perform with higher F0 and energy. It indicates that fine-grained accent intensity changes can be easily detected. The Ai-TTS system provides an effective explicit accent intensity control mechanism. We suggest the readers to access our online website [4] for more demos.

3.5. Accent Expression

To understand how the accent renderer performs, we randomly select 100 utterances from the test set as the test samples and report the 5-scale Mean Opinion Score (MOS) for three systems, including **Ground Truth** L2 speech, synthesized L2 speech by **FastSpeech2** [18] and our **Ai-TTS**. For fair comparison, we set i to 5 for all input utterances of Ai-TTS. We invite 20 listeners and report the subjective MOS results in the second column of Table 1. It is observed that our Ai-TTS achieves a MOS of 4.01 ± 0.022, that is significantly higher than *FastSpeech2* baseline and very close to the *Ground Truth*. For objective evaluation, we follow [18] and report the moments (including standard deviation (σ), skewness (γ) and kurtosis ($\mathcal{K}$)), and average dynamic time warping (DTW) [36] (ϱ) of the pitch distribution between the synthesized L2-accented speech and the ground truth reference in the third to sixth columns of Table 1. It can be seen that the Ai-TTS system is reported with all values that are closer to those of the Ground Truth than FastSpeech2. The subjective and objective evaluations suggest that our Ai-TTS with accent renderer achieves more expressive L2 speech in terms of accent expression.

4. Conclusion

We have studied a novel TTS model, named Ai-TTS, to control the L2 accent and its intensity explicitly. We have conducted a series of experiments on utterance-and phoneme-level intensity control to validate the effectiveness of the Ai-TTS model. The proposed GoP based intensity score outperforms the adversarial weight strategy in terms of interpretability and controllability. This work marks an important step towards controllable rendering of accented TTS synthesis. In future work, we plan to further improve the intensity quantification method.

[4] Speech samples: `https://ttslr.github.io/Ai-TTS/`

5. References

[1] Y. Zhang, R. J. Weiss, H. Zen, Y. Wu, Z. Chen, R. Skerry-Ryan, Y. Jia, A. Rosenberg, and B. Ramabhadran, "Learning to speak fluently in a foreign language: Multilingual speech synthesis and cross-language voice cloning," *Proc. Interspeech 2019*, pp. 2080–2084, 2019.

[2] L. Loots and T. Niesler, "Automatic conversion between pronunciations of different english accents," *Speech Communication*, vol. 53, no. 1, pp. 75–84, 2011.

[3] C. Veaux, J. Yamagishi, and S. King, "The voice bank corpus: Design, collection and data analysis of a large regional accent speech database," *Proc. O-COCOSDA 2013*, pp. 1–4, 2013.

[4] M. L. García Lecumberri, R. Barra Chicote, R. Pérez Ramón, J. Yamagishi, and M. Cooke, "Generating segmental foreign accent," *Proc. Interspeech 2014*, pp. 1303–1306, 2014.

[5] M. Waseem and C. Sujatha, "Speech synthesis system for indian accent using festvox," *International journal of Scientific Engineering and Technology Research, ISSN*, pp. 2319–8885, 2014.

[6] S. Kayte, M. Mundada, and D. C. Kayte, "Speech synthesis system for marathi accent using festvox," *International Journal of Computer Applications*, vol. 130, no. 6, pp. 38–42, 2015.

[7] G. E. Henter, J. Lorenzo-Trueba, X. Wang, M. Kondo, and J. Yamagishi, "Cyborg speech: Deep multilingual speech synthesis for generating segmental foreign accent with natural prosody," *Proc. ICASSP 2018*, pp. 4799–4803, 2018.

[8] Z. Liu and B. Mak, "Multi-lingual multi-speaker text-to-speech synthesis for voice cloning with online speaker enrollment." *Proc. Interspeech 2020*, pp. 2932–2936, 2020.

[9] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan *et al.*, "Natural tts synthesis by conditioning wavenet on mel spectrogram predictions," *Proc. ICASSP 2018*, pp. 4779–4783, 2018.

[10] M. J. Munro, T. M. Derwing, and S. L. Morton, "The mutual intelligibility of l2 speech," *Studies in second language acquisition*, vol. 28, no. 1, pp. 111–131, 2006.

[11] R. Pérez-Ramón, M. L. García Lecumberri, and M. Cooke, "Foreign accent strength and intelligibility at the segmental level," *Speech Communication*, vol. 137, pp. 70–76, 2022.

[12] C. Wutiwiwatchai, A. Thangthai, A. Chotimongkol, C. Hansakunbuntheung, and N. Thatphithakkul, "Accent level adjustment in bilingual thai-english text-to-speech synthesis," in *2011 IEEE Workshop on Automatic Speech Recognition & Understanding*. IEEE, 2011, pp. 295–299.

[13] R. Liu, B. Sisman, G. Gao, and H. Li, "Expressive tts training with frame and style reconstruction loss," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 1806–1818, 2021.

[14] R. Liu, B. Sisman, G. l. Gao, and H. Li, "Decoding knowledge transfer for neural text-to-speech training," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, pp. 1–1, 2022.

[15] R. Liu, B. Sisman, G. Gao, and H. Li, "Expressive tts training with frame and style reconstruction loss," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 1806–1818, 2021.

[16] R. Liu, B. Sisman, and H. Li, "Graphspeech: Syntax-aware graph attention network for neural speech synthesis," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 6059–6063.

[17] Y. Wang, R. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio *et al.*, "Tacotron: A fully end-to-end text-to-speech synthesis model," *Proc. Interspeech 2017*, pp. 4006–4010, 2017.

[18] Y. Ren, C. Hu, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu, "Fastspeech 2: Fast and high-quality end-to-end text to speech," in *International Conference on Learning Representations*, 2020.

[19] R. Liu, B. Sisman, Y. Lin, and H. Li, "Fasttalker: A neural text-to-speech architecture with shallow and group autoregression," *Neural Networks*, vol. 141, pp. 306–314, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0893608021001532

[20] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," *The journal of machine learning research*, vol. 17, no. 1, pp. 2096–2030, 2016.

[21] H. Franco, H. Bratt, R. Rossier, V. Rao Gadde, E. Shriberg, V. Abrash, and K. Precoda, "Eduspeak®: A speech recognition and pronunciation scoring toolkit for computer-aided language learning applications," *Language Testing*, vol. 27, no. 3, pp. 401–418, 2010.

[22] L. Yang, J. Zhang, and T. Shinozaki, "Self-supervised learning with multi-target contrastive coding for non-native acoustic modeling of mispronunciation verification," *Proc. Interspeech 2022*, pp. 4312–4316, 2022.

[23] S. M. Witt, "Automatic error detection in pronunciation training: Where we are and where we need to go," in *International Symposium on automatic detection on errors in pronunciation training*, 2012, pp. 1–8.

[24] S. M. Witt and S. J. Young, "Phone-level pronunciation scoring and assessment for interactive language learning," *Speech communication*, vol. 30, no. 2-3, pp. 95–108, 2000.

[25] J. Kong, J. Kim, and J. Bae, "Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis," *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[26] A. Waibel, "Consonant recognition by modular construction of large phonemic time-delay neural networks," *Advances in neural information processing systems*, vol. 1, 1988.

[27] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.

[28] V. Peddinti, D. Povey, and S. Khudanpur, "A time delay neural network architecture for efficient modeling of long temporal contexts," *Proc. Interspeech 2015*, 2015.

[29] W. Hu, Y. Qian, F. K. Soong, and Y. Wang, "Improved mispronunciation detection with deep neural network trained acoustic models and transfer learning based logistic regression classifiers," *Speech Communication*, vol. 67, pp. 154–166, 2015.

[30] Z. Meng, H. Hu, J. Li, C. Liu, Y. Huang, Y. Gong, and C.-H. Lee, "L-vector: Neural label embedding for domain adaptation," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7389–7393.

[31] R. Liu, B. Sisman, B. W. Schuller, G. Gao, and H. Li, "Accurate emotion strength assessment for seen and unseen speech based on data-driven deep learning," *Proc. Interspeech 2022*, pp. 5493–5497, 2022.

[32] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," *Proc. ICASSP 2015*, pp. 5206–5210, 2015.

[33] G. Zhao, S. Sonsaat, A. Silpachai, I. Lucic, E. Chukharev-Hudilainen, J. Levis, and R. Gutierrez-Osuna, "L2-arctic: A non-native english speech corpus," *Proc. Interspeech 2018*, pp. 2783–2787, 2018.

[34] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[35] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[36] M. Müller, "Dynamic time warping," *Information retrieval for music and motion*, pp. 69–84, 2007.