Adaptive End-to-End Text-to-Speech Synthesis Based on Error Correction Feedback from Humans

Kazuki Fujii, Yuki Saito and Hiroshi Saruwatari
Graduate School of Information Science and Technology, The University of Tokyo,
7-3-1 Hongo Bunkyo-ku, Tokyo 133-8656, Japan
E-mail: kazuki-fujii@g.ecc.u-tokyo.ac.jp, {yuuki_saito, hiroshi_saruwatari}@ipc.i.u-tokyo.ac.jp

***Abstract*—We propose an end-to-end text-to-speech (TTS) method that can intuitively correct accent errors in synthetic speech by using feedback from humans. State-of-the-art end-to-end TTS methods can synthesize high-quality speech, but humans can hardly interpret the black-boxed TTS system represented as a stack of neural networks. This reduced interpretability prevents humans from controlling the TTS system to correct errors in synthetic speech intuitively. In this paper, we focus on a method to involve human listeners in the process of accent error correction for synthetic speech generated by end-to-end TTS. Specifically, we build an end-to-end TTS model equipping a prosody predictor that estimates the change of pitch for each syllable from phoneme embeddings and context-aware word embedding. Then, we perform a human-in-the-loop (HITL) framework to correct errors of the prosody predictor using collective intelligence of human listeners. The results of Japanese TTS experiments show that our HITL framework can successfully correct accent errors and contribute to the quality of synthetic speech comparable to the conventional method requiring an accent dictionary for text analysis.**

I. Introduction

Text-to-speech (TTS) [1] is a technology that uses text as input to synthesize the corresponding intelligible and natural speech. End-to-end TTS [2] is a method for synthesizing a speech waveform by feeding text into a single deep neural network (DNN)-based TTS model instead of pipelined process utilized in statistical parametric speech synthesis (SPSS) [3]. Since expertise in TTS is no longer necessarily required, end-to-end TTS enables the users to synthesize naturally-sounding speech close to human speech [4]. However, the end-to-end TTS users often suffer from the low interpretability of the entire TTS model represented by a stack of DNNs when they correct errors of synthetic speech to obtain their desired outcomes. One primal error is an accent error of synthetic speech, which can degrade the quality of synthetic speech and can even impede accurate communication. Therefore, a framework is needed to enable users of TTS, including non-experts, to correct errors. In this paper, we focus on accent errors of synthetic speech generated by end-to-end TTS and investigate a method to involve human listeners in the error correction process.

One approach for improving the controllability of speech synthesized by end-to-end TTS is to estimate prosodic information of the speech from the input text and condition the TTS model by the information explicitly. Such prosodic information should be interpretable for users to intuitively correct errors in synthetic speech. For instance, Kurihara et al.'s method [5] inserts symbols for the prosody control of synthetic speech into the input phoneme sequence. However, some of the symbols are complicated for non-expert users who want to correct accent errors. In addition, if the text analysis result is incorrect, the user, who may not be an expert, needs to update the dictionary used for the analysis to ensure that the error does not happen again. In summary, although this conventional method can provide an intuitive way to control the prosody of synthetic speech, its adaptability, i.e., the ability to make the error correction easy, has not yet been examined. The improvement of adaptability will cultivate an advanced society where humans and computers can use speech as a natural means of communication with each other.

In this paper, we propose an end-to-end TTS method that can easily correct accent errors in synthetic speech by using feedback from humans. First, we build an end-to-end TTS model equipping a DNN-based prosody predictor that estimates the change of pitch for each syllable (raising, lowering, or keeping unchanged) from phoneme embeddings and context-aware word embeddings. The simplification of symbols proposed by Kurihara et al. makes the error correction more intuitive for non-expert users because they only have to modify the prosody predictor output for each syllable from one of the three options to control the prosody of synthetic speech. Then, we perfrorm a human-in-the-loop (HITL) framework involving human listeners to collect accent annotations of synthetic speech and to tell a trained end-to-end TTS model how it can improve the prosody prediction performance. The results of Japanese TTS experiments show that our HITL framework can successfully correct accent errors and achieve the quality of synthetic speech comparable to the conventional method requiring an accent dictionary for text analysis.

II. Related Work

The prediction of natural prosody is an essential and challenging task in end-to-end TTS [2] because a single DNN must learn the one-to-many mapping from phoneme/character sequence to multiple voices caused by variation in non-/para-linguistic information. If the predicted prosody is incorrect, it can degrade the quality of synthetic speech and cause

miscommunication, especially in TTS for a tonal language such as Japanese and Chinese. For instance, "箸" (chopsticks), "橋" (bridge), and "端" (edge) are all pronounced as "*hashi*" in Japanese, but all with different accents to distinguish their meanings. In this section, we briefly review conventional studies related to modeling and controlling prosody for end-to-end TTS.

One straightforward approach for improving the prosody prediction performance of end-to-end TTS is to introduce techniques and features used in traditional SPSS methods [3]. For example, Okamoto et al. [6] used a full-context label including rich linguistic features derived from text analysis (e.g., part of speech and accent type) as the input of end-to-end TTS. Ren et al. [7] proposed FastSpeech 2, a non-autoregressive end-to-end TTS model that incorporates the predictions of phoneme durations and acoustic features (F0 and energy) into the process of mel-spectrogram generation from a phoneme sequence. Our work follows these approaches partially but aims to improve the interpretability of linguistic features to control synthetic speech because non-expert users can hardly understand the role of each feature correctly.

Several studies on end-to-end TTS have aimed to improve the controllability of synthetic speech, another vital factor in some applications using TTS technology as the primary means of communication, such as spoken dialogue systems [8] and speaking aids [9]. Some presented machine learning frameworks to learn discrete representations for prosody control in an unsupervised manner using deep generative models [10] (e.g., variational autoencoders [11]). Others designed more interpretable features for the input of end-to-end TTS, such as phonetic and prosodic (PP) labels [5] and intuitive prosodic features [12]. The PP labels deeply inspire our proposed method because they can provide an intuitive way to control the prosody of synthetic speech through a series of symbols: "^" (initial raising), "!" (accent nucleus), "#" (accentual phrase boundary), "(" (declarative end-of-sentence), "?" (interrogative end-of-sentence), and "_" (pause). However, some symbols require the user's knowledge of speech and linguistics to understand correctly, which makes the error correction of prosody difficult and reduces *adaptability* of synthetic speech.

III. Proposed Method

We propose a method to improve both controllability and adaptability of end-to-end TTS. In this paper, we describe our system designed for the Japanese language. Fig. 1 shows an overview of the proposed method.

A. Baseline TTS model

We use FastSpeech 2 [7] as the baseline TTS model of our proposed method because we place importance on the speed of learning and inference as well as the stability. The original FastSpeech 2 consists of some modules for generating a mel-spectrogram of speech from a phoneme sequence:

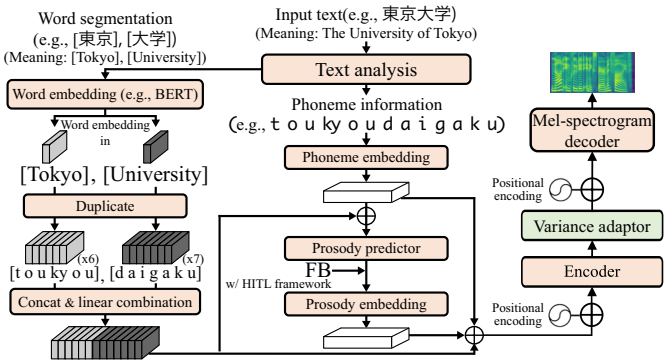

Fig. 1. Overview of proposed method. The prosody predictor predicts the prosody symbol from phoneme embeddings and context-aware word embeddings derived from BERT.

1) Text analysis for converting the input raw text into phoneme sequences.
2) Phoneme embedding to obtain an embedding vector of phoneme identities.
3) Encoder for producing a hidden phoneme sequence from the embedding vector.
4) Variance adaptor for prediction of variance information of speech, such as duration, pitch, and energy, from the hidden sequence and adding the information to it.
5) Mel-spectrogram decoder for predicting mel-spectrogram from the variance adaptor outputs.

B. Prosody predictor

For improving the adaptability of synthetic speech, we introduce a prosody predictor, a DNN that estimates the pitch change for each syllable, into the original FastSpeech 2. The DNN architecture is almost the same as each variance adaptor in the original FastSpeech 2. Specifically, it has a 2-layer 1D convolutional network with ReLU activation [13], each followed by a layer normalization [14], a dropout [15] layer, and a linear layer for projecting the hidden states onto the output sequence.

1) Inputs—Phoneme and context-aware word embeddings: In addition to phoneme embeddings, we use word embeddings derived from BERT [16] as input to the prosody predictor for considering the contextual information of input text. The process from input raw text to the prosody predictor is as follows. First, BERT and its tokenizer are used to segment the text by word to obtain a word embedding for each word. In this process, word embedding is obtained by taking the average of multiple embeddings of subwords that constitute the word. Then, each word embedding is duplicated as many times as the number of phonemes in the word to align the different time resolutions of phoneme/word embeddings. Finally, these embeddings are summed and used as input to the prosody predictor.

2) Outputs—Interpretable prosody symbols: We define the output of the prosody predictor as a simplified version of the PP labels used in Kurihara et al.'s method: 1) "[" (raising the

TABLE I
DEFINITIONS OF PROSODY SYMBOLS.

Symbol	Definition	Prosody predictor output
[	Raising the pitch	$+1$
]	Lowering the pitch	-1
_	Keeping the accent unchanged	0

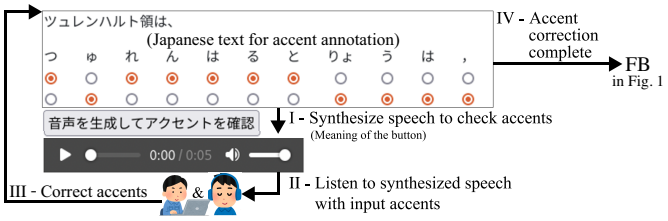

Fig. 2. Interface and overview of accent error correction feedback.

pitch), 2) "]" (lowering the pitch), and 3) "_" (keeping the accent unchanged). The simplification is based on the fact that each syllable (i.e., mora) in a Japanese text has a high/low accent pattern and is expected to provide an accessible way for users to manipulate the prosody of synthetic speech.

3) Loss function: We use the mean squared error (MSE) between predicted and ground-truth prosody symbols for the prosody predictor training. The ground-truth prosody symbols can be obtained in various ways: 1) results of the same text analysis as in Kurihara et al.'s method, 2) unsupervisedly learned latent variables of prosody (e.g., Yufune et al.'s method [10]), and 3) accent correction feedback from humans (explained later in Section III-C). We use the first way in this paper for simplicity. However, we can also use the second or third in situations where the accent dictionary is unavailable to obtain the ground-truth symbols, such as dialect TTS [17]. In the computation of the loss function, we convert each prosody symbol into a scalar value as shown in Table I to enable the loss computation as MSE. Since all components in the prosody predictor are described by DNNs, we can perform the backpropagation algorithm to update the predictor's model parameters.

C. Human-in-the-loop accent error correction

1) Motivation: The HITL framework aims to improve the adaptability of synthetic speech by involving human listeners in the accent error correction process. The core idea is to make a human listener serve as *teacher* who tells correct accent annotations on the synthetic speech, presumably containing some errors, to the TTS model.

2) Accent error correction and its interface: Fig. 2 shows the interface and overview of our HITL framework. The interface consists of five components: 1) a target text and its corresponding mora sequences, 2) radio buttons to annotate a high/low accent for each mora (hiragana), 3) a button to synthesize speech with the annotated accent patterns, 4) human listeners and 5) a voice playback button to listen to the synthesized speech. In our framework, human listeners annotate accent patterns of the text by clicking either of the upper/lower radio buttons for each mora. To improve the reliability of annotation results, we allow the listeners to confirm the results of their annotations by playing back synthetic speech generated from our model with the annotated accents and continue to modify the annotations if needed. After this "annotation and synthesis" HITL finishes, we can obtain multiple candidates for the accent pattern of the target text from the listeners.

3) Feedback aggregation: A simple way to reflect the accent annotation feedback in our TTS model is to choose one from all annotation results and to use it for synthesizing speech. However, the ability of accent annotation differs from listener to listener, and some listeners can provide feedback that even worsen the quality of synthetic speech. Therefore, we aggregate the multiple annotation results corrected through our HITL framework using the following methods.

- **Mode**: Taking the mode of high/low accents for each mora.
- **Multi annotator competence estimation (MACE) [18]**: Estimating the unknown answers for annotations and the annotator's ability, based on the expectation-maximization algorithm. As a result, the outcomes of listeners with low ability are excluded from the feedback aggregation.

IV. EXPERIMENTS

A. Conditions for basic TTS

We used the JSUT corpus [19], which consists of Japanese speech by a female speaker. The speech data was downsampled to 22,050 Hz, and the dimension of the mel-spectrogram was 80. For F0 analysis, the WORLD vocoder [20] was used. The number of training and evaluation data were 4,488 and 512 sentences in the BASIC5000 subset of the JSUT corpus, respectively. We used the implementation of FastSpeech 2 provided on GitHub[1]. The phoneme alignment was obtained by Julius [21]. We used tdmelodic [22] as a dictionary for the text analysis lexicon to enable the estimation of accurate accents for a variety of texts. The dimensions of all embeddings were set to 256, and the optimizer for DNN training was Adam [23] with a learning rate of 0.001. The neural vocoder was HiFi-GAN [24] pretrained and published as UNIVERSAL_V1[2]. For the BERT model, we used bert-base-japanese-v2[3]. In addition, the dimension of the word embedding was compressed from 768 to 256 by a linear projection layer.

The following four methods were compared.

- **FS2**: FastSpeech 2 trained without conditioning on any prosody symbols.

[1]https://github.com/Wataru-Nakata/FastSpeech2-JSUT
[2]https://github.com/jik876/hifi-gan
[3]https://huggingface.co/cl-tohoku/bert-base-japanese-v2

- **FS2+Symbols**: FastSpeech 2 trained by conditioning on the interpretable prosody symbols directly. First, a text and its corresponding prosody symbols are input into FastSpeech 2. Next, the text is converted to phoneme embeddings and the prosody symbols are converted to prosodic embeddings. Finally, those elements and word embeddings are summed and input to FastSpeech 2's encoder for learning and inference.
- **FS2+Predictor**: The proposed method described in Section III.
- **FS2+Predictor (target)**: The DNN architecture and its training procedure were the same as the FS2+Predictor but the ground-truth prosody symbols (i.e., derived from the text analysis using an accent dictionary) were used as the input.

To investigate the effect of the prosody predictor, we add the method FS2+Symbols method, which directly conditions FastSpeech 2 by the prosody symbols. The number of training steps for each TTS model was 100,000.

B. Conditions for HITL accent error correction

Experiments on the HITL framework for accent error correction in TTS were conducted with crowdworkers recruited through the crowdsourcing platform Lancers[4]. To assess whether our HITL framework can effectively collect accent error correction feedback, we used the VOICEACTRESS100 (VA100), a subset of the JSUT corpus including many texts (100 sentences) whose accents are difficult to estimate, such as proper nouns and coined words. The accents for each sentence in the VA100 subset were annotated by 15 different crowdworkers, i.e., $15 \times 100 = 1500$ crowdworkers in this experiment. The 15 collected prosody symbol sequences per sentence were then aggregated into a single sequence using "Mode" or "MACE". In addition, we synthesized speech using all sequences and took the root mean squared error (RMSE) of the logarithmic fundamental frequency (logF0) between the synthesized and natural speech, and chose samples based on the logF0 values: the lowest (Best), in the center (Median), and the highest (Worst).

C. Objective evaluation

We conducted an objective evaluation to examine the prosody prediction accuracy of the newly introduced prosody predictor. In order to focus only on pitch prediction accuracy, we used the duration of natural speech for TTS in this experiment, and we used the logF0 RMSE [cent] as the evaluation criterion.

Fig. 3 shows a violin plot of the objective evaluation results that compare the performances of four methods. As we can see, FS2+Predictor performs worse than FS2+Symbols, but FS2+Predictor (target) is comparable to FS2+Symbols. These results indicate that 1) errors in the newly introduced prosody predictor significantly degrade the prosody prediction performance, but 2) the degraded performance can be improved if the ground-truth prosody symbols are available.

[4]https://www.lancers.jp/

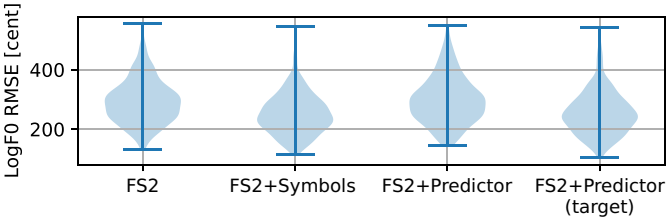

Fig. 3. Objective evaluation results of four compared methods. We used 512 utterances in BASIC5000 subset calculate logF0 RMSE.

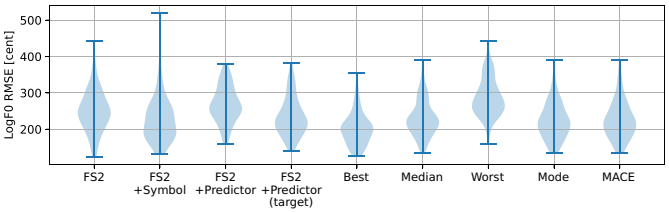

Fig. 4. Results of objective evaluation of four compared methods and our HITL framework. We used 100 utterances in VA100 subset to calculate logF0 RMSE.

Fig. 4 shows the objective evaluation results that examine the effectiveness of our HITL framework. From this figure, FS2+Predictor is worse than FS2 or FS2+Symbols. However, the upper outliers in FS2 and FS2+Symbols tend to improve by introducing the prosody predictor. One of the causes might be the effect of training the prosody predictor together with the TTS model. Focusing on the results of our HITL framework (Best, Median, and Worst), there are non-negligible differences in the ability of crowdworkers to annotate accents of synthetic speech. Furthermore, the two aggregation methods, Mode and MACE, achieve the logF0 comparable to that of FS2+Predictor (target). These results indicate that our HITL framework can collect accent annotations sufficient to synthesize speech produced by using the text-analysis-derived prosody symbols.

D. Subjective evaluation

We conducted a subjective evaluation of the prosodic naturalness to determine whether the tendency shown in the objective evaluation is similar to that in the human evaluation. All subjective evaluations were conducted using Lancers.

1) Evaluation results of four TTS methods: We first conducted a series of preference AB tests to compare all combinations of the four TTS methods except for the FS2+Predictor and FS2+Predictor (target) pair. In the method-pairwise test, each listener evaluated 10 pairs of speech samples synthesized by a specific method-pair. The number of listeners for each AB test was 25, i.e., $25 \times 5 = 125$ listeners participated in the evaluation.

Table II shows the results of preference AB tests. As we can see, the FS2+Predictor (target) score is better than that

TABLE II
PREFERENCE SCORES FOR PROSODY NATURALNESS OF SYNTHETIC SPEECH. **BOLD** DENOTES A SIGNIFICANT DIFFERENCE BETWEEN THE TWO METHODS ($p < 0.05$).

Compared method	Preference score
FS2 vs. FS2+Predictor	**0.552 vs. 0.448**
FS2 vs. FS2+Predictor (target)	**0.268 vs. 0.732**
FS2+Symbols vs. FS2+Predictor	**0.768 vs. 0.232**
FS2+Symbols vs. FS2+Predictor (target)	0.520 vs. 0.480
FS2 vs. FS2+Symbols	**0.248 vs. 0.752**

of FS2+Predictor. The lack of significant difference between FS2+Symbols and FS2+Predictor (target) indicates that the newly introduced prosody predictor can synthesize speech with natural prosody if the prosody prediction is correct.

Then, we conducted a mean opinion score (MOS) test to compare the four methods and the ground-truth JSUT speaker's voices regarding the naturalness of speech prosody. We prepared four speech samples for each method in the MOS test and presented them to listeners in random order. The listeners rated the naturalness of each sample on a 5-point scale (1: very poor–5: very good). The number of listeners was 50, and each listened to 20 speech samples.

Table III shows the MOS test results. The tendencies of the results are similar to those obtained in the objective evaluation, i.e., FS2+Symbols and FS2+Predictor (target) are significantly better than the others and the two methods are comparable. These results demonstrate that our simplified prosody symbols are sufficient in improving the naturalness of prosody in synthetic speech.

2) Evaluation results of HITL accent error correction: Finally, we conducted a MOS test to verify the effectiveness of our HITL framework. We compared the same methods as shown in Fig. 4 except for FS2 and FS2+Symbols because we focused on the correctness of obtained accent annotation feedback in the proposed method. The number of listeners was 50, and each evaluated 28 speech samples.

Table IV shows the MOS test results. The results of FS2+Predictor and Worst are significantly lower than the others, which indicates that 1) prediction errors of the newly introduced prosody predictor can degrade the speech naturalness and 2) the choice of annotation results substantially affect the naturalness. Meanwhile, Mode and MACE achieve MOSs significantly higher than that of FS2+Predictor, indicating that our HITL framework can successfully compensate for accent errors in end-to-end TTS and contribute to the quality improvement of synthetic speech. In addition, there is no significant difference between scores of Mode and MACE, which suggest that the majority of the crowdworkers had high ability and the effect of crowdworkers with low ability was minor.

V. CONCLUSION

In this paper, we proposed an end-to-end TTS method that can easily correct accent errors in synthetic speech based on human listeners' feedback. We presented an HITL framework

TABLE III
MOS IN TERMS OF NATURALNESS OF SYNTHETIC SPEECH (TTS MODEL EVALUATION). **BOLD** VALUE IS COMPARABLE TO THAT OF FS2+SYMBOLS ($p > 0.05$).

Method	MOS $\pm$ 95% confidence interval
JSUT	4.24 ± 0.139
FS2+Predictor	2.76 ± 0.143
FS2+Predictor (target)	$\mathbf{3.35 \pm 0.163}$
FS2+Symbols	3.45 ± 0.158
FS2	2.71 ± 0.157

TABLE IV
MOS IN TERMS OF NATURALNESS OF SYNTHETIC SPEECH (HITL EXPERIMENT EVALUATION). **BOLD** VALUES ARE HIGHER THAN THOSE OF FS2+PREDICTOR ($p < 0.05$).

Method	MOS $\pm$ 95% confidence interval
FS2+Predictor	2.87 ± 0.163
FS2+Predictor (target)	$\mathbf{3.54 \pm 0.156}$
Best	$\mathbf{3.62 \pm 0.151}$
Median	$\mathbf{3.38 \pm 0.156}$
Worst	2.84 ± 0.174
Mode	$\mathbf{3.63 \pm 0.159}$
MACE	$\mathbf{3.49 \pm 0.161}$

that corrects accent errors using collective intelligence. The experimental results show that the proposed HITL framework works well in correcting errors in the synthesized speech and contributes to quality improvement. As future work, we will conduct further studies on the user interface of feedback and how to integrate the obtained prosodic sequences.

ACKNOWLEDGMENT

This work was supported by JST, Moonshot R&D Grant Number JPMJPS2011.

REFERENCES

[1] Y. Sagisaka, "Speech synthesis by rule using an optimal selection of non-uniform synthesis units," in *Proc. ICASSP*, New York, U.S.A., Apr. 1988.

[2] Y. Wang, R. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio, Q. Le, Y. Agiomyrgiannakis, R. Clark, and R. A. Saurous, "Tacotron: Towards end-to-end speech synthesis," in *Proc. INTERSPEECH*, Stockholm, Sweden, Aug. 2017, pp. 4006–4010.

[3] H. Zen, K. Tokuda, and A. Black, "Statistical parametric speech synthesis," *Speech Communication*, vol. 51, no. 11, pp. 1039–1064, Nov. 2009.

[4] J. Shen, R. Pang, R. J. 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*, Calgary, Canada, Apr. 2018, pp. 4779–4783.

[5] K. Kurihara, N. Seiyama, and T. Kumano, "Prosodic features control by symbols as input of sequence-to-sequence acoustic modeling for neural TTS," *IEICE Transactions on Information and Systems*, vol. E104.D, no. 2, pp. 302–311, Feb. 2021.

[6] T. Okamoto, T. Toda, Y. Shiga, and H. Kawai, "Real-time neural text-to-speech with sequence-to-sequence acoustic model and WaveGlow or single gaussian WaveRNN vocoders," in *Proc. INTERSPEECH*, Graz, Austria, Sep. 2019, pp. 1308–1312.

[7] 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 *Proc. ICLR*, Vienna, Austria, May 2021.

[8] D. F. Glas, T. Minato, C. T. Ishi, T. Kawahara, and H. Ishiguro, "ERICA: The ERATO intelligent conversational android," in *Proc. RO-MAN*, New York, U.S.A., Aug. 2016, pp. 22–29.
[9] Y. Oshima, S. Takamichi, T. Toda, G. Neubig, S. Sakti, and S. Nakamura, "Non-native text-to-speech preserving speaker individuality based on partial correction of prosodic and phonetic characteristics," *IEICE Transactions on Information and Systems*, vol. E99.D, no. 12, pp. 3132–3139, 2016.
[10] K. Yufune, T. Koriyama, S. Takamichi, and H. Saruwatari, "Accent modeling of low-resourced dialect in pitch accent language using variational autoencoder," in *Proc. SSW*, Budapest, Hungary, Aug. 2021, pp. 189–194.
[11] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv*, vol. abs/1312.6114, 2013.
[12] T. Raitio, R. Rasipuram, and D. Castellani, "Controllable Neural Text-to-Speech Synthesis Using Intuitive Prosodic Features," in *Proc. INTERSPEECH*, Shanghai, China, 2020, pp. 4432–4436.
[13] A. F. Agarap, "Deep learning using rectified linear units (ReLU)," *arXiv*, vol. abs/1803.08375, 2018.
[14] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv*, vol. abs/1607.06450, 2016.
[15] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, Apr. 2014.
[16] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," in *Proc. NAACL-HLT*, Minneapolis, USA, June 2019, pp. 4171–4186.
[17] M. Pucher, D. Schabus, J. Yamagishi, F. Neubarth, and V. Strom, "Modeling and interpolation of austrian german and viennese dialect in hmm-based speech synthesis," *Speech Communication*, vol. 52, no. 2, pp. 164–179, 2010.
[18] D. Hovy, T. Berg-Kirkpatrick, A. Vaswani, and E. Hovy, "Learning whom to trust with MACE," in *Proc. NAACL-HLT*, 2013, pp. 1120–1130.
[19] S. Takamichi, R. Sonobe, K. Mitsui, Y. Saito, T. Koriyama, N. Tanji, and H. Saruwatari, "JSUT and JVS: Free Japanese voice corpora for accelerating speech synthesis research," *Acoustical Science and Technology*, vol. 41, no. 5, pp. 761–768, Sep. 2020.
[20] M. Morise, F. Yokomori, and K. Ozawa, "WORLD: a vocoder-based high-quality speech synthesis system for real-time applications," *IEICE Transactions on information and systems*, vol. E99-D, no. 7, pp. 1877–1884, Jul. 2016.
[21] A. Lee, T. Kawahara, and K. Shikano, "Julius — an open source real-time large vocabulary recognition engine," in *Proc. EUROSPEECH*, Aalborg, Denmark, Sep. 2001, pp. 1691–1694.
[22] H. Tachibana and Y. Katayama, "Accent estimation of Japanese words from their surfaces and romanizations for building large vocabulary accent dictionaries," in *Proc. ICASSP*, May 2020, pp. 8059–8063.
[23] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. ICLR*, San Diego, California, U.S.A., May 2015.
[24] J. Kong, J. Kim, and J. Bae, "HiFi-GAN: Generative adversarial networks for efficient and high fidelity speech synthesis," in *Proc. NeurIPS*, vol. 33, Virtual Conference, Dec. 2020, pp. 17 022–17 033.