Investigation of Japanese PnG BERT Language Model in Text-to-Speech Synthesis for Pitch Accent Language

Yusuke Yasuda, *Member, IEEE*, and Tomoki Toda, *Senior Member, IEEE*

***Abstract*—End-to-end text-to-speech synthesis (TTS) can generate highly natural synthetic speech from raw text. However, rendering the correct pitch accents is still a challenging problem for end-to-end TTS. To tackle the challenge of rendering correct pitch accent in Japanese end-to-end TTS, we adopt PnG BERT, a self-supervised pretrained model in the character and phoneme domain for TTS. We investigate the effects of features captured by PnG BERT on Japanese TTS by modifying the fine-tuning condition to determine the conditions helpful inferring pitch accents. We manipulate content of PnG BERT features from being text-oriented to speech-oriented by changing the number of fine-tuned layers during TTS. In addition, we teach PnG BERT pitch accent information by fine-tuning with tone prediction as an additional downstream task. Our experimental results show that the features of PnG BERT captured by pretraining contain information helpful inferring pitch accent, and PnG BERT outperforms baseline Tacotron on accent correctness in a listening test.**

***Index Terms*—PnG BERT, text-to-speech, Japanese, pitch accent, self-supervised learning.**

I. Introduction

END-TO-END (E2E) text-to-speech synthesis (TTS) can generate highly natural synthetic speech from raw texts [1], [2]. However, rendering correct pitch accents or tones, which are accents involving pitch change, remains a challenging problem for E2E-TTS [3]–[5]. The accuracy of rendering pitch accents is crucial in pitch accent languages such as Japanese, because pitch accents control the meaning of words. Conventional TTS systems resolve accent information with a morphological analyzer-based text front-end by looking it up in a accent dictionary, which is normally expensive and requires language-specific knowledge to construct [6]. The characteristics of E2E-TTS to use texts or phonemes directly as input enables it to be applied to various data and languages without constructing a pronunciation dictionary. On the other hand, the challenge of rendering a pitch accent in an E2E-TTS approach comes from building knowledge corresponding to accent dictionary implicitly from text and speech pair.

To tackle the challenge of rendering the correct pitch accent in Japanese E2E-TTS, two main issues should be addressed. One is the low word coverage in the speech corpus. Pitch accents in pitch accent languages are mainly determined by words, so a low word coverage in training speech data can result in the main problem of poor pitch accents for an end-to-end TTS model. However, even a large-scale speech corpus can never match a lexicon dictionary in terms of word coverage [7]. The other problem is the diversity of characters. The Japanese writing system contains ideographic characters that represent meaning rather than pronunciation. Owing to the diversity of ideographic characters, Japanese end-to-end TTS from raw texts has been impossible. Therefore, phonemes are used as input in end-to-end Japanese TTS instead. This results in loss of word-related information, leading to incorrect pitch accents [3], [5]. Both phonemes and pitch accents are important speech representations to render in Japanese TTS, but these two representations depend on different features: rendering phonemes depends on characters and phonemes, and rendering pitch accents depends on word and syntactic features. Therefore, a framework to capture both surface form and high-level features is required.

To tackle these two main issues, we utilize a self-supervised pre-trained model using a large-scale text corpus. As a self-supervised learning method, we use PnG BERT [8]. PnG BERT is an extension of BERT [9] designed for TTS as a downstream task by capturing both word and phoneme contexts. We adopt PnG BERT for Japanese end-to-end TTS because (1) it can capture both word and phoneme contexts; (2) word and phoneme alignment can be learned in text domain instead of the speech domain; (3) a downstream task can be performed against features corresponding to a phoneme segment instead of a word segment. Regarding (1), we can expect that features extracted from PnG BERT are helpful to render correct pitch accents, because Japanese pitch accents are mainly determined by words. Moreover, BERT can not only provide semantic information but also syntactic information [10], [11]. Syntactic information such as conjugation type is also helpful inferring the fundamental frequency (F_o) in Japanese speech [12]. Concerning (2), introducing multiple soft-attention layers to TTS enables to us align multiple linguistic units to speech [13] and to overcome the

Manuscript received 14 January 2022; revised 24 April 2022 and 22 June 2022; accepted 2 July 2022. Date of publication 13 July 2022; date of current version 14 October 2022. This work was supported in part by commissioned by NEDO under Project JPNP20006. The guest editor coordinating the review of this manuscript and approving it for publication was Dr. Hung-yi Lee. *(Corresponding author: Yusuke Yasuda.)*

The authors are with the Information Technology Center, Nagoya University, Nagoya City 4648-601, Japan (e-mail: yasuda.yusuke@g.sp.m.is.nagoya-u.ac.jp; tomoki@icts.nagoya-u.ac.jp).

This work involved human subjects or animals in its research. The author(s) confirm(s) that all human/animal subject research procedures and protocols are exempt from review board approval.

Digital Object Identifier 10.1109/JSTSP.2022.3190672

lack of word boundary symbols in the Japanese writing system. However, it is preferable to align multiple language units in the text domain considering the limited volume of clean speech data. On the other hand, PnG BERT can learn alignment between words and phonemes in self-attention layers [14] by utilizing large-scale text corpus [8]. This characteristic enables feature extraction considering both words and phonemes for TTS. With (3), we can introduce tone prediction as a downstream task of PnG BERT in addition to TTS straightforwardly. Tone representation of Japanese pitch accent is determined in at the syllable level. Therefore the tone can be predicted in the phoneme part of PnG BERT token-by-token. Naturally, we can expect that tone prediction is also benefited from word or syntactic information captured by PnG BERT considering (1). The tone prediction is used to explicitly teach PnG BERT accent information in this work.

In this study, we investigate the effects of feature contents captured by PnG BERT on Japanese TTS by modifying the fine-tuning condition. In our experiments, we manipulate the content of PnG BERT features from being text-oriented to speech-oriented by changing the number of fine-tuned layers by TTS. In addition, we inject pitch accent information itself to PnG BERT features by fine-tuning with tone prediction as an additional downstream task.

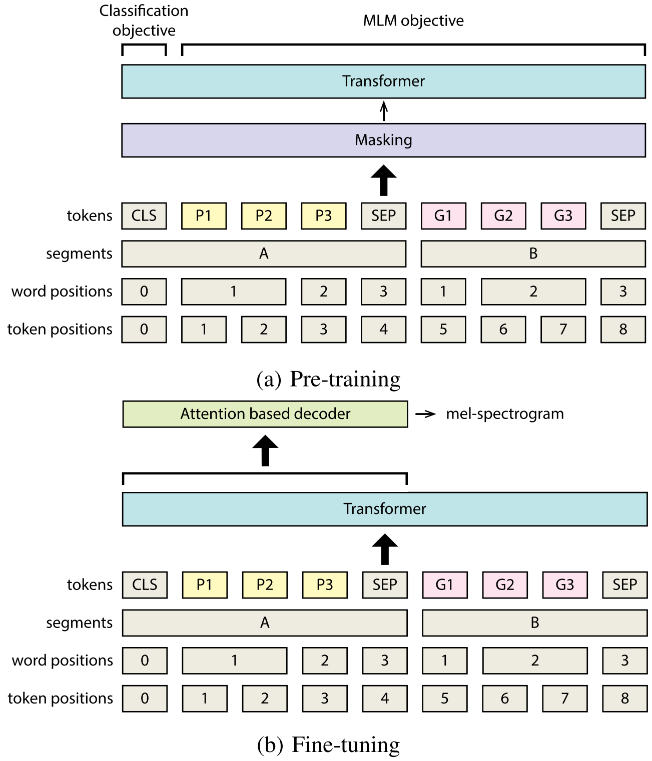
This paper is organized as follows. In Section II, we describe backgrounds of BERT, PnG BERT and background of Japanese TTS. In Section III we introduce our proposed method for Japanese TTS using PnG BERT. Section IV shows our experimental results. In Section V, we summarize related works. Finally, In Section VI, we conclude our findings.

II. Background

A. BERT

BERT [9] is a self-supervised learning-based language model for general language representation. It has two steps to learn textual representation and task-related features: pre-training and fine-tuning. BERT uses the masked language model (MLM) objective during pre-training to capture general language representation. The MLM objective optimizes a model by predicting masked tokens given input texts in which some parts of the tokens are randomly masked. In addition to MLM, BERT has the next sentence prediction (NSP) objective, which classifies whether two sentences are adjacent or unrelated. During fine-tuning, BERT uses a supervised objective to solve a specific task. Examples of a downstream task to fine-tune BERT are natural language processing (NLP) tasks such as language understanding and question answering.

To use the MLM and NSP objectives, BERT arranges input tokens as follows: an input sequence always starts with a `CLS` token, which is used for a classification task such as NSP. Each input sentence ends with a `SEP` token, which means the end of a sentence. As for the masking method, a portion (e.g., 15%) of the input tokens are randomly chosen to be predicted as targets in self-supervised learning. Most of the selected tokens (e.g., 80%) are masked by being replaced with a `MASK` token Some tokens (e.g., 10%) are replaced with a random token, whereas some tokens (e.g., 10%) are not replaced. A BERT model is trained to recover the original sentences from the modified inputs.



(a) Pre-training

(b) Fine-tuning

Fig. 1. Mechanism of PnG BERT. P: phoneme, G: grapheme.

The architecture of BERT is equivalent to that of the encoder of Transformer [14]. It consists of a stack of self-attention blocks that are a combination of a multihead self-attention layer and a feed-forward network with ReLU activation. Positional encoding is used to take the order of tokens into consideration.

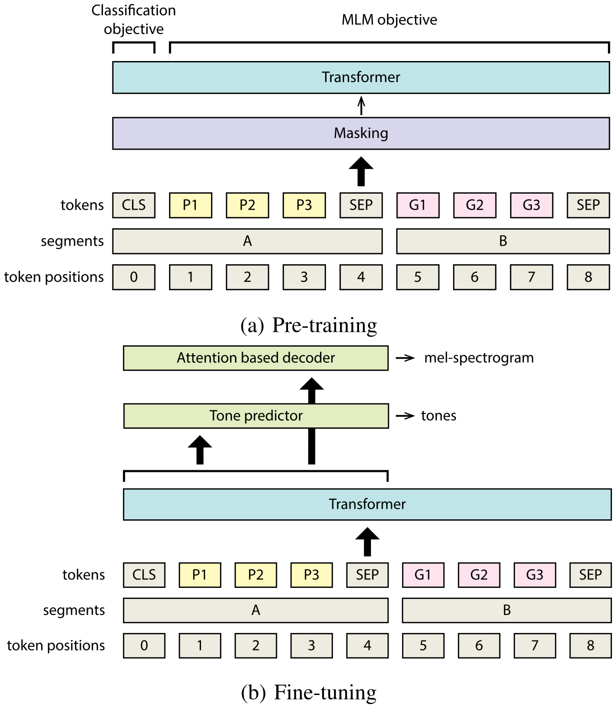
B. Png BERT

Language models for TTS tasks aim to capture pronunciation-related information, whereas language models for NLP tasks focus on syntactic or semantic information. PnG BERT is a self-supervised learning-based language model that can capture both contextual word and phoneme information in the text domain [8]. PnG BERT is an extended method of BERT: it uses the same Transformer-based architecture and MLM training objective. PnG BERT extends BERT in terms of input representation, masking strategy, and fine-tuning to capture pronunciation information.

Fig. 1 shows the structure of PnG BERT. It uses graphemes and phonemes as input representation instead of words. Here, we use graphemes to refer to as any textual representation unit such as characters and subwords. A target downstream task of PnG BERT is TTS. Character and phonemes are common input representations for end-to-end TTS. The input sequence for PnG BERT consists of the concatenation of phoneme and grapheme sequences of a corresponding text. Each of the phoneme and grapheme sequences ends with a `SEP` token. A `CLS` symbol is prepended to the whole sequence. The `CLS` token is for the sentence classification task. Token, word, and segment positions are specified by positional encoding [14], because self-attention in Transformer is permutation invariant.

The masking strategy of PnG BERT is designed at the word level. Because the input representations in PnG BERT are graphemes and phonemes, masking at token-level limits contextual information captured in pretraining up to surface-form level. The word-level masking enables PnG BERT to learn all layers of contextual information: semantic, surface form, and pronunciation. In PnG BERT, one of the following masking strategies is chosen for a random word: both graphemes and phonemes are masked; only graphemes are masked; only phonemes are masked; both graphemes and phonemes are kept intact; both graphemes and phonemes are randomly replaced with other phonemes and graphemes. For evaluation, grapheme-to-phoneme (G2P) and phoneme-to-grapheme (P2G) masking strategies may be used in addition to MLM. In the G2P masking, all the phoneme segments are masked. In P2G masking, all the grapheme segments are masked.

A TTS downstream task can be conducted with PnG BERT by feeding its final outputs corresponding to the phoneme segment into the TTS decoder. PnG BERT models are jointly fine-tuned with the TTS decoder while freezing its parameters in part of its layers to prevent the loss of learned knowledge during pre-training. PnG BERT is compatible with any TTS decoder.



(a) Pre-training

(b) Fine-tuning

Fig. 2. Mechanism of Japanese PnG BERT to perform pre-training and fine-tuning with text-to-speech synthesis and tone prediction as downstream task. P: phoneme, G: grapheme.

C. Japanese TTS

Japanese is classified as a pitch accent language in terms of prosodic description. Japanese pitch accents are lexical, which means that accents depend on words or a combination of words to form an accentual phrase. The Japanese pitch accent has an accentual nucleus position that is specified by a pitch fall within an accent phrase. The accent nucleus position is measured in a syllable unit called mora. Although the pitch accent of individual words can be described in a lexicon, it is affected by adjacent words in an accentual phrase, which is a phenomenon called accent sandhi.

The Japanese TTS system uses text front-end for word segmentation and lexicon lookup for pronunciation by using a morphological analyzer. After the lexicon lookup, accent sandhi is estimated on the basis of rules [15] or machine learning [16], [17]. The Japanese text front-end provides full-context labels that include various type of information helpful to TTS such as part of speech. Ref. [18] shows that at least accent nucleus position, or the accentual type, should be provided to TTS to render a natural pitch accent.

III. Japanese TTS Using PnG BERT

Fig. 2 shows our method to conduct Japanese TTS using PnG BERT. Its general framework is the same as that of the plain PnG BERT: graphemes and phonemes are concatenated as one sequence, and the sequence is fed to Transformer layers; the PnG BERT model is pretrained with the MLM objective by masking a grapheme and phoneme pair at the word level; during fine-tuning and speech generation, the phoneme part of the output of the last Transformer layer is used as the input for a TTS decoder to generate a spectrogram.

We introduce two changes to the PnG BERT to conduct Japanese TTS. The first modification is the absence of word-level alignment information between graphemes and phonemes as word positions in the inputs. The Japanese writing system lacks clear word boundary symbols. To obtain word boundaries of Japanese texts used for pre-training, we use morphological analyzers. Obviously, automatically derived word boundaries from the morphological analyzer contain errors. We do not use word positions to extract features to avoid dependence on erroneous information. Thus, word-level alignment information is required only during pre-training to conduct word-level masking. We expect that removing word positions does not affect the performance of TTS negatively as Ref. [8] suggests. We also expect that masking based on erroneous word boundaries derived from the morphological analyzer does not have major negative effects, because it just affects size of word units but does not affect consistency of word alignments between graphemes and phonemes. It is known that the consistency of the word-level masking affects the performance of pre-training and the downstream TTS task in PnG BERT [8]. We will confirm these assumptions by measuring the G2P conversion performance of PnG BERT.

The second modification is the introduction of tone prediction to TTS as an additional downstream task. Japanese pitch accents are an important spoken feature to disambiguate homonyms. Unfortunately, accent labels are not available on the scale of large corpora that are used for pre-training. Therefore, we incorporate tone prediction as one of the downstream tasks. The tone prediction task is performed against the phoneme part of PnG BERT's output features similar to the TTS downstream task. We use tone labels to represent accent information because they are aligned to phonemes in a syllable unit. Ideally, the relationship between extracted features and pitch accents is to be learned implicitly from the target mel-spectrogram that contains F_o information. The tone prediction task is to explicitly teach PnG BERT pitch accents. We expect that fine-tuning PnG BERT with accent prediction makes it easy for PnG BERT to learn pitch accents.

IV. Experimental Evaluations

To investigate the effects of feature contents captured by PnG BERT on Japanese TTS, we pre-trained a Japanese PnG BERT model and constructed TTS models under various fine-tuning conditions.

A. Pre-Training Condition

To construct a PnG BERT model, we used a BERT model with model size, so-called BERT-base [9], which consists of 12 Transformer layers with 768 hidden size.[1] The model is trained up to 11.63 M steps with 20 batch size and learning rate with linear decay started from 5.0×10^{-5} with the Adam optimizer with L2 regularization [19]. Note that the original PnG BERT uses the SM3 optimizer [20] instead of Adam. We found that with Adam, it was easy to optimize the model, whereas with SM3, it was difficult to find the proper hyper parameters, although it was memory-efficient.

We randomly selected 25% words in a sentence as a target of the MLM objective. For the 25% selected words, we applied the following masking strategy: (a) both graphemes and phonemes are masked for 48% random words; (b) only the character part is masked for 8% random words; (c) only the phoneme part is masked for 8% random words; (d) both characters and phonemes are kept intact for 10% random words; (d) both characters and phonemes are randomly replaced with other phonemes or characters for 10% random words. Note that (b) and (c) are not present in the original PnG BERT [8]. In preliminary experiment, we found that these masking strategies were very effective for improving G2P and P2G performances.

We followed the input representations of the original PnG BERT paper [8]. We used character and phoneme representations as input to PnG BERT. The two sequences were appended with the `SEP` token and concatenated into one sequence. The `CLS` token was prepended, but we did not use the NSP objective.

As Japanese text corpora for pre-training, we used Wikipedia[2] and Aozorabunko.[3] Table I shows statistics of the text corpora used in this study. Wikipedia is a collection of online encyclopedia. We used a subset of Japanese Wikipedia that contained about 18.5 M sentences. Aozorabunko is collection of Japanese books in public the domain. We used subset of Aozorabunko that contained about 4.9 M sentences. Phoneme transcriptions were obtained by morphological analysis using the Kuromoji analyzer[4] with Neologd dictionary [21]–[23]. For validation, we used text and phoneme labels from the JSUT corpus [24], which was also used to fine-tune a TTS model. We used a subset of JSUT corpus called `basic5000`, which contains 5,000 sentences that cover the pronunciations of commonly used ideographic characters.

TABLE I
Statistics of Japanese Text Corpus Used in This Study

Corpus	#Records	Usage	
		Pre-training	Fine-tuning
`Wikipedia`	18,486,538	Training	-
`Aozorabunko`	4,891,288	Training	-
`JSUT`	6,385	Validation	Training/Validation/Test

B. TTS Systems

We constructed five TTS systems using the same pre-trained PnG BERT model on the basis of different fine-tuning conditions.

1) `PGB0`: No fine-tuning
2) `PGB2`: Two-layer fine-tuning
3) `PGB4`: Four-layer fine-tuning
4) `PGB6`: Six-layer fine-tuning
5) `PGB2T`: Two-layer fine-tuning with tone prediction in addition to TTS
6) `PGB2MC`: Two-layer fine-tuning and all graphemes are masked during fine-tuning and prediction.

The `PGB0` system is equivalent to TTS using fixed features extracted from PnG BERT. The `PGB2`, `PGB4`, and `PGB6` systems conducted fine-tuning of the last two, four, and six layers while freezing the other layers, respectively. The `PGB2T` uses multi-downstream tasks of both TTS and tone prediction. In all systems, only an output from the last layer was used as input feature to TTS. Note that the `PGB2T` system did not use tone labels during prediction. `PGB2MC` was to distinguish the contribution of the graphemes by masking all graphemes during fine-tuning and prediction.

We constructed another pre-trained BERT model that uses only phonemes during pre-training, fine-tuning and prediction as a baseline to clarify the contribution of the graphemes. In this system, pre-training, fine-tuning, and prediction scheme were same as PnG BERT except that all graphemes were masked out. We trained phoneme BERT up to 1.0 M steps with 80 batch size with linear decay started from 5.0×10^{-5} with the Adam optimizer with L2 regularization. We fine-tuned the last two layers of phoneme BERT with TTS task. We referred to this system as `PB2MC`.

7) `PB2MC`: Phoneme BERT that uses only phonemes as inputs during pre-training and fine-tuning. The last two layers of BERT encoder are fine-tuned.

We prepared five baseline systems with different encoder and input representations.

[1] We also tried the original model size, which consists of 6 Transformer layers with 512 hidden size. We found that fine-tuning of PnG BERT with the original model size converged faster and its synthetic speech sounded similar quality as PnG BERT with BERT-base size.

[2] [Online]. Available: https://dumps.wikimedia.org/

[3] [Online]. Available: https://github.com/aozorabunko/aozorabunko/

[4] [Online]. Available: https://github.com/atilika/kuromoji

8) PGBN: Non-pretrained PnG BERT as the encoder with grapheme and phoneme labels
9) TAC: Tacotron2's encoder with phoneme labels
10) TACT: Tacotron2's encoder with phoneme and tone labels
11) BTAC2A: Tacotron2's phoneme encoder and pre-trained character BERT encoder. Its two sources are aligned to target with dual source attention [25].
12) BTAC2B: Tacotron2's phoneme encoder and pre-trained character BERT encoder. Its two sources are aligned to target with dual source biattention [26].

The PGBN system used the same framework to conduct TTS with PnG BERT except for skipping pre-training. It used the same BERT architecture, input representation, and TTS procedure as the other five systems using PnG BERT.

We used Tacotron2 [1] as the baseline method, different from PnG BERT. We constructed two Tacotron2 models without pre-training: TAC that uses the phoneme label and TACT that uses phoneme and tone labels as inputs. The phoneme and tones were concatenated after being processed by the pre-net layer to feed to the encoder in the TACT system [27]. We used the same phoneme and tone labels that were used in the fine-tuning of TTS and tone prediction for the PnG BERT-based systems. The encoder of Tacotron2 consists of three convolutional layers and a bidirectional LSTM layer. Note that TAC is a baseline comparable to the PnG BERT systems considering that the PnG BERT systems did not use tone labels during prediction, and TACT worked as the upper bound as it used tone labels during prediction.

We also included two Tacotron baseline systems extended with pre-trained character BERT encoder. These systems combines the character BERT encoder and phoneme encoder by dual source attention [25]. This was a common method to combine BERT with Tacotron [13], [28], and we referred to the system as BTAC2A. This method did not consider alignment between the two sources, i.e. phoneme features and character features. The BTAC2B system used dual source biattention to learn alignment between the two sources [26]. With biattention, the model learned to align the two sources, and then the aligned sources are aligned to targets to predict outputs. Guided attention loss [29] were used for alignments between BERT features and target mel-spectrogram to enforce monotonicity. We used a publicly available pre-trained Japanese BERT model that used characters as input representation and was trained with word-level masking.[5] We fine-tuned the last two layers of BERT encoder.

All these systems used the same decoder as that of Tacotron2 [1]: two-layer LSTM decoder, attention layer with forward attention [27], and two-layer pre-net bottleneck. The forward attention enabled faster training by enforcing a monotonic structure of alignment [27]. The decoder pre-net layer prevented the decoder from excessively depending on autoregressive feedback by applying dropout [30].

C. Fine-Tuning Conditions

1) Fine-Tuning With TTS Task: The TTS models were fine-tuned with a batch size of 60 and learning rate of 1.0×10^{-6} until validation loss stopped improving. The PGB0 system was trained up to 488 K steps by using pre-trained PnG BERT parameters. The PGB2 system was trained up to 272 K steps by warm-starting from parameters of PGB0. The PGB4 system was trained up to 90 K steps by warm-starting from parameters of PGB2. The PGB6 system was trained up to 142 K steps by warm-starting from parameters of PGB4. The PGBN system was trained up to 282 K steps from scratch. The PGB2MC system was fine-tuned up to 72 k steps by warm-starting from parameters of PGB2. The PB2MC system was fine-tuned up to 156 k steps by freezing all layers and then was fine-tuned up to 32 k steps by unfreezing the last two layers. The BTAC2A system was fine-tuned up to 80 k steps by freezing all layers and then was fine-tuned up to 20 k steps by unfreezing the last two layers. The BTAC2B system was fine-tuned up to 64 k steps by freezing all layers and then was fine-tuned up to 26 k steps by unfreezing the last two layers. We used the same optimizer, i.e., Adam with L2 regularization, to fine-tune a TTS model [19].

We extracted an 80-dimensional mel-spectrogram from a waveform with a 24 k sampling rate as a target acoustic feature. We used Parallel Wave GAN [31] to generate a waveform from the mel-spectrogram.

We used JSUT [24] as the speech corpus to fine-tune TTS models. We used phoneme labels transcribed from speech.[6] We split the data into train, validation, and test sets with 5,386, 499, and 500 sentences, respectively.

2) Fine-Tuning With TTS and Tone Prediction Task: To capture accent information, the PnG BERT model was fine-tuned by the tone prediction task in the PGB2T system. In this setting, a PnG BERT model predicts tone labels from the phoneme segment of an input sequence.

The PGB2T system was trained up to 180 K steps by warm-starting from parameters of PGB0 with the Adam optimizer with L2 regularization. The fine-tuning of tone prediction was performed at the same time with TTS fine-tuning. We found that the multitask learning of TTS and tone prediction was essential to incorporate tone prediction to PnG BERT. Separating the fine-tuning by TTS and tone prediction tasks in two stages failed to learn the alignment between input features from PnG BERT and target speech during TTS training.

We used the same corpus as that of TTS, namely, JSUT corpus [24] to fine-tune with tone prediction.[7] The data split setting for tone prediction was the same as that of TTS. We used the X-JToBI [32] format to represent the tone labels. The tone label contains the following pitch contour patterns: L (neutral low), and H (neutral high), %L (neutral low at the start of accentual phrase), L% (neutral low at the end of accentual phrase), and

[5] BERT-base_mecab-ipadic-char-4k_do-whole-word-mask. [Online]. Available: https://github.com/cl-tohoku/bert-japanese

[6] kana_level3. [Online]. Available: https://github.com/sarulab-speech/jsut-label

[7] The accent labels. [Online]. Available: https://github.com/sarulab-speech/jsut-label

A (accent nucleus position). The tone labels were aligned to phoneme labels.

D. Evaluation Methods

1) Pre-Training Evaluation: We evaluated the performance of pre-training with three metrics: accuracies of MLM, G2P, and P2G as in [8]. The accuracy of MLM was computed on the basis of the MLM objective with random masking, the same as in the pre-training. The accuracy of G2P was computed by masking the entire phoneme segment, and the accuracy of P2G was computed by masking the entire grapheme segment.

2) Objective Evaluations: We measured attention error rate (AER) [18], character error rate (CER), tone prediction accuracy (TA), phrase prediction accuracy (PA), and accentual nucleus prediction accuracy (AA) as objective metrics.

The AER was used to evaluate how easily the feature from PnG BERT was aligned to speech. The AER was calculated on the basis of the number of non-monotonic attention distributions that contain discontinuities or overestimation of duration. We counted discontinuities over four encoder time steps and duration over 30 decoder time steps as alignment errors.

The CER was used to evaluate the phonetic correctness of synthetic speech. The CER refers to the character-level Levenshtein edit distance between reference texts and hypothesis texts transcribed by automatic speech recognition (ASR). We used ESPNet [33] for ASR .[8]

The TA, PA, and AA were used to evaluate the abundance of tone-related information in encoded features. The TA is prediction accuracy of all tone labels described in Section IV-C2. The PA is the prediction accuracy of start and end of accentual phrase (%L and L%). The AA is the prediction accuracy of accent nucleus (A). They were measured by conducting linear classification against fixed features from the encoder of PnG BERT or Tacotron. In the case of the BTAC2A system, concatenation of outputs from phoneme encoder and BERT encoder were used as inputs for the tone classifier. In the case of the BTAC2B system, concatenation of outputs from phoneme encoder and aligned BERT encoder by biattention were used as inputs for the tone classifier.

3) Subjective Evaluations: We conducted a listening test to evaluate the synthetic speech.[9] We included the ten systems described in Section IV-B except for PGB2MC, PB2MC, BTAC2A, and BTAC2B in addition to natural samples (NAT) and analysis-by-synthesis (ABS) in the listening test. Here, ABS meant synthetic samples from neural vocoder given ground truth mel-spectrogram. We asked two questions to listeners. The first question was about naturalness in the five-scale mean opinion score (MOS): very bad, bad, acceptable, good, and very good. The second question was about accent correctness in four-scale MOS: totally wrong, somewhat wrong, somewhat correct, and totally correct. Every sample was evaluated five times. We collected 25,000 evaluations in total from 248 Japanese listeners. We checked the statistical significance with the Mann-Whitney rank test [34].

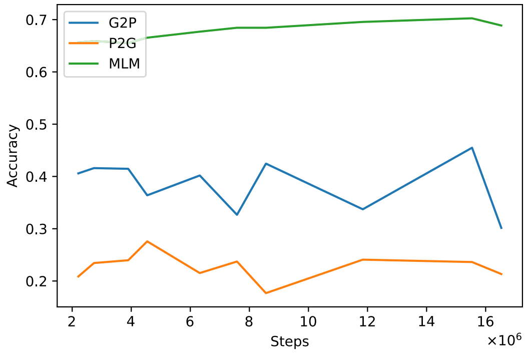


Fig. 3. History of pre-training accuracy of a PnG BERT model for validation set. MLM refers to accuracy under random masking as training time. G2P refers to accuracy under masking all phonemes. P2G refers to accuracy under masking all graphemes.

TABLE II
Alignment Error Rate (AER), Character Error Rate (CER), Tone Prediction Accuracy (TA), Phrase Prediction Accuracy (PA), and Accentual Nucleus Prediction Accuracy (AA) of TTS Systems. Pred. Denotes Prediction

System	Pre-training	Fine-tuning	Tone	AER↓	CER↓	TA↑	PA↑	AA↑
PGB0	✓	-	-	13.6	29.4	72.8	74.1	20.6
PGB2	✓	2 layers	-	5.2	19.7	73.1	73.5	22.6
PGB4	✓	4 layers	-	4.2	19.0	72.9	73.0	22.4
PGB6	✓	6 layers	-	3.6	18.6	70.0	66.8	16.2
PGBN	-	All	-	13.2	44.2	58.4	27.8	26.0
PGB2T	✓	2 layers	Pred.	7.6	23.0	81.6	80.5	56.7
PGB2MC	✓	2 layers	-	6.8	25.5	66.5	56.5	23.4
PB2MC	✓	2 layers	-	5.6	28.8	64.1	38.3	27.0
TAC	-	All	-	4.4	21.0	66.0	56.2	20.9
TACT	-	All	Label	5.4	19.9	88.5	82.7	69.0
BTAC2A	✓	2 layers	-	3.0	21.8	65.1	49.9	11.4
BTAC2B	✓	2 layers	-	20.6	22.0	60.7	48.2	17.1
ABS				-	21.0	-	-	-
NAT				-	24.9	-	-	-

E. Experimental Results

1) Pre-Training Results: Fig. 3 shows the accuracies of MLM, G2P, and P2G in the validation set as the performance of pre-training. The accuracy of MLM consistently improved during pre-training up to 70.3% until overfit. On the other hand, the accuracies of G2P and P2G fluctuated during pre-training, which were 45.5% and 23.6% at the model with the highest MLM accuracy, respectively. It seemed that the performances of G2P and P2G had a trade-off relationship: when one metric increased, the other decreased. The magnitudes of these values were somewhat consistent with [8]. The relatively low performances of G2P and P2G indicated that the features captured by PnG BERT were not dominant in surface-form information with our masking strategy of pre-training, which was expected to contain syntactic and semantic information [8].

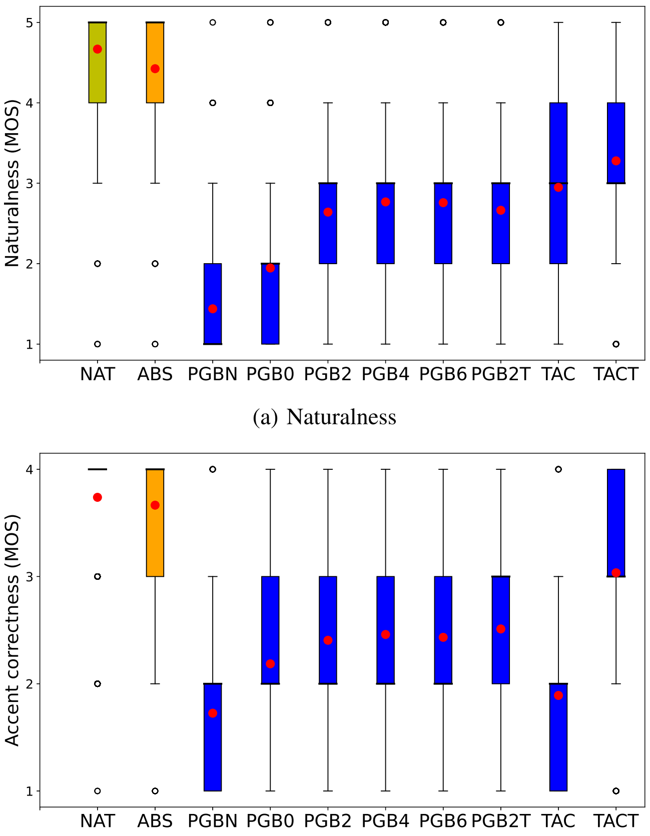
2) TTS Results: Table II shows AER, CER, TA, PA, and AA from the TTS systems. The AER and CER showed roughly the same trends: the systems that had high AER also showed high

[8] We used a publicly available ASR model. [Online]. Available: https://zenodo.org/record/4304245#.Yac84FORVqs

[9] Our audio sample page. [Online]. Available: https://todalab.github.io/yasuda-japanese-pngbert-samples/

CER. When the number of fine-tuning layers in PnG BERT was increased, both AER and CER improved, which indicated that features from PnG BERT contained information more aligned to speech. The high AER and CER from the PGB0 system showed that the pre-trained PnG BERT without fine-tuning provides abundant textual features and less spoken features, and it was consistent with the relatively low G2P accuracy in the pre-training. The PGBN system showed high AER and CER although it fine-tuned all layers in PnG BERT. This implied that pre-training was essential for PnG BERT to learn stable alignment between phonemes and speech for TTS.

As for tone prediction accuracy, pre-trained PnG BERT systems (PGB0-6) showed high accuracies. They showed high PA but low AA, which indicated that the high TA values mainly came from rich accentual phrase information in the encoded features, and accentual nucleus information was not sufficiently captured in the features. The system with tone prediction (PGB2T) had high AA, which indicating that its feature contained both accentual phrase and nucleus information. The masking graphemes (PGBMC) resulted in degradation of TA from 73.1% to 63.3%. The degradation was mainly caused by significant decrease of PA from 73.5% to 38.4%, suggesting that phrase information depends largely on contextual information of graphemes rather than phonemes. The low TA and PA of the PnG BERT system without pre-training (PGBN) indicated that pre-training was essential to capture information related to accentual phrase. Tacotron, which did not perform pre-training, also failed to capture sufficient information related to accentual phrase, as indicated by its low TA and PA. The high tone prediction accuracy of TACT was because tone labels were given as input. Both PnG BERT systems that use only phonemes (PGB2MC and PB2MC) showed low TA compared to the system using both characters and phonemes (PGB2). The performance drop of TA was mainly caused by degradation of PA. This indicated that contextual information encoded in character part enriched phrase-related information. TA from these systems were similar values to that of the Tacotron baselines. This suggested that pre-training or fine-tuning with only phonemes did not help to encode tone-related information. Both Tacotron systems extended with character BERT (BTAC2A and BTAC2B) showed similar values of TA, which were lower compared to TA from PnG BERT systems. We thought that the no improvement of TA from BERT incorporated Tacotron systems were caused by poor alignments between phoneme features and BERT features. We found that it was hard to learn alignment between BERT features and mel-spectrogram in both BTAC2A and BTAC2B systems, and alignment between BERT features and phoneme features in the BTAC2B system even though various techniques such as forward attention, guided attention, and biattention were used to enforce robust alignments. The attention distributions about BERT features were mostly blurred or nonmonotonic. These results were consistent with the existing works [13], [28]. We thought learning alignment proprly between phoneme features and BERT features was important to predict tones from the contextual features because tone labels were designed in phoneme-level, and high TA from PnG BERT systems came from learning accurate alignments between phonemes and characters by utilizing large-scale text resources during pre-training.



(a) Naturalness

(b) Accent correctness

Fig. 4. Results of listening test.

Fig. 4 shows results of the listening test on (a) naturalness and (b) accent correctness. It showed that pre-training was crucial, considering the significant improvement of naturalness from 1.44 ± 0.02 (PGBN) to 1.95 ± 0.03 (PGB0). Fine-tuning was also essential, considering the significant improvement of naturalness from 1.95 ± 0.03 (PGB0) to 2.64 ± 0.03 (PGB2). Increasing the number of fine-tuned layers slightly improved naturalness from 2.64 ± 0.03 (PGB2) to 2.77 ± 0.03 (PGB4), but there was no significant difference between the four and six layers, indicating that four layers were sufficient for fine-tune. The tone prediction did not help in improving naturalness, because there was no significant difference between PGB2 and PGB2T. The PnG BERT-based systems did not match the naturalness of the Tacotron that had MOS of 2.95 ± 0.03 (TAC). We found that samples from the PnG BERT-based systems had lower fidelity than samples from the Tacotron systems. We consider that the low fidelity was caused by the difficulty of optimizing the fine-tuning of PnG BERT, as indicated by long fine-tuning time requirements and relatively high loss values. The Tacotron with accent labels (TACT) had the highest MOS score of 3.28 ± 0.03 among TTS systems as expected of upper bound system.

For accent correctness, pre-training and fine-tuning were crucial as well. There was no significant difference among systems with different numbers of fine-tuned layers. The tone prediction slightly improved accent correctness from 2.41 ± 0.03 (PGB2) to 2.51 ± 0.03 (PGB2T). The pre-trained PnG BERT based systems significantly outperformed Tacotron, which had MOS of 1.89 ± 0.03 (TAC), in terms of accent correctness in contrast to the results on naturalness. This suggests that pre-trained features from PnG BERT were helpful in inferring pitch accents. The PnG BERT without pre-training, which showed a low score of 1.73 ± 0.03 (PGBN), contained graphemes in inputs, so what helped in inferring pitch accents was not surface form information but presumably syntactic and semantic information. Still, accents inferred by the pre-trained PnG BERT-based systems were not sufficiently accurate compared to accent labels, as indicated by their lower scores than TACT, which used accent labels during prediction. The Tacotron with accent labels (TACT) showed relatively high MOS score of 3.04 ± 0.03, which indicated that Japanese listeners were sensitive to pitch accent. We consider that the word coverage of training speech and tone label data used for fine-tuning limited the correctness of inferred accents, because the only phase in which accent nucleus positions can be learned by PnG BERT was fine-tuning in our method. Considering that accent nucleus positions are the spoken features of words, we think that pre-training with tone labels or fine-tuning with large-scale speech data would be required to improve the accent correctness of Japanese PnG BERT.

V. Related Works

An early attempt to utilize an unsupervised textual representation for TTS is the vector space model, which is an information retrieval method to derive linguistic feature vectors from texts by singular value decomposition [35]. The objective of the vector space model is to improve the language versatility of TTS by replacing text front-end, which is traditionally dependent on many language-specific features. However, the vector space model is not helpful for DNN-based TTS [36].

Neural-network-based pre-training methods replace the vector space model by using much larger text resources. The neural-network-based pre-training is useful for TTS in ideographic languages to overcome data sparsity caused by character diversity. For example, pre-trained character embedding is used for a front-end model in Maindarin TTS [37]. Pre-trained linguistic encoder is used to enable to take ideographic characters directly as input in Mandarin TTS by introducing G2P objective [38].

Word embedding [39], [40] and contextual word embedding such as BERT [9] are widely used unsupervised word representation learning methods for general purposes. Word embedding is mainly used to improve prosody of synthetic speech in TTS [28], [41]–[46]. Moreover, the word embedding is suitable for prosody modeling in variational autoencoder (VAE)-based TTS [47]–[49]. Word embedding is also used to improve data efficiency under small amounts of training speech data [50].

BERT can not only be used to extract fixed contextual word embedding features, but can also be customized to fit the downstream task. There are three strategies to customize a BERT model: (1) fine-tuning, (2) input representation, and (3) masking strategy. Fine-tuning enables the fitting of features to a downstream task by tuning part of the parameters of the pre-trained model. It is known that fine-tuning improves the performance of BERT in downstream tasks similar to pre-training methods in the NLP domain [51]. Recent TTS works using BERT perform fine-tuning [8], [42], [46], [48], [49]. Some studies show that the fine-tuning of BERT with TTS improves the naturalness of synthetic speech [8], [46], [48]. In this study, we fine-tuned PnG BERT with TTS and tone prediction and confirmed the effectiveness of fine-tuning.

The design of the input representation of BERT for TTS includes words, subwords, characters, phonemes, and speech. Pre-training in the word domain obtains semantic information, and pre-training in the character domain obtains surface information, and pre-training in the subword domain is considered between them. Words [45], [45] and subwords [42], [46]–[49] are commonly used representations for BERT. In [44], characters were used as input to BERT for Mandarin TTS. PnG BERT [8] combines both characters and phonemes to pre-train BERT to capture both surface and phoneme information to apply to TTS. This idea to combine subwords and phonemes is also investigated in spoken language understanding [52]. In [53], mel-spectrogram was used to pre-train BERT, which is used as a pre-net network [30] in TTS.

The masking strategy can be designed specifically for a downstream task. PnG BERT [8] applies word-level masking on both character and phoneme pairs. [52] uses "one-mode masking" was used on subwords and phonemes, which is to mask the entire subword or phoneme segment equivalent to the G2P or P2G task.

VI. Conclusion

In this work, we investigated the effects of features captured by PnG BERT on Japanese TTS by modifying the fine-tuning condition to determine the conditions helpful in rendering pitch accents. We manipulated the content of PnG BERT features from being text-oriented to speech-oriented by changing the number of fine-tuned layers during TTS. In addition, we taught PnG BERT pitch accent information by fine-tuning with tone prediction as an additional downstream task. Our experiment showed that pre-training and fine-tuning were essential for both naturalness and accent correctness for Japanese TTS. Increasing the number of fine-tuned layers improved alignment and character error rates, but did not contribute considerably to naturalness. PnG BERT provided better accent correctness than the Tacotron baseline, although all the PnG BERT-based systems were inferior to Tacotron in terms of naturalness because of their low fidelity. Fine-tuning with tone prediction considerably improved tone prediction accuracies from captured features, but its improvements were limited in the subjective evaluation of accent correctness. Overall, the features of PnG BERT captured by pre-training contained information helpful in inferring pitch accent, and fine-tuning by TTS enriched pitch accent information for the PnG BERT features.

Our future works include the improvement of the low fidelity of generated speech and long fine-tuning time of PnG BERT. It would be interesting to find the optimal masking strategy to obtain higher fidelity and fine-tuning efficiency of Japanese PnG BERT by changing the masking strategy to control the balance of surface form, syntactic, and semantic information. We will also work on the further improvement of Japanese PnG BERT to infer pitch accents. One interesting way to improve the capture of pitch accent features is pre-training PnG BERT with accent labels instead of fine-tuning by tone prediction. Another interesting way is to fine-tune PnG BERT with large-scale speech data.

REFERENCES

[1] J. Shen et al., "Natural TTS synthesis by conditioning WaveNet on MEL spectrogram predictions," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2018, pp. 4779–4783.

[2] N. Li, S. Liu, Y. Liu, S. Zhao, M. Liu, and M. Zhou, "Close to Human quality tts with transformer," 2018, *arXiv:1809.08895v3*.

[3] Y. Yasuda, X. Wang, S. Takaki, and J. Yamagishi, "Investigation of enhanced tacotron text-to-speech synthesis systems with self-attention for pitch accent language," in *Proc. IEEE Int. Conf. Acoust.*, *Speech Signal Process.*, 2019, pp. 6905–6909.

[4] B. Li, Y. Zhang, T. Sainath, Y. Wu, and W. Chan, "Bytes are all you need: End-to-end multilingual speech recognition and synthesis with bytes," in *Proc. IEEE Int. Conf. Acoust.*, *Speech Signal Process.*, 2019, pp. 5621–5625.

[5] T. Fujimoto, K. Hashimoto, K. Oura, Y. Nankaku, and K. Tokuda, "Impacts of input linguistic feature representation on Japanese end-to-end speech synthesis," in *Proc. 10th ISCA Workshop Speech Synth.*, 2019, pp. 166–171.

[6] The HTS Working Group, "The Japanese TTS system 'Open JTalk'," 2015. [Online]. Available: http://open-jtalk.sourceforge.net/

[7] J. Taylor and K. Richmond, "Analysis of pronunciation learning in end-to-end speech synthesis," in *Proc. Interspeech*, 2019, pp. 2070–2074.

[8] Y. Jia, H. Zen, J. Shen, Y. Zhang, and Y. Wu, "PnG BERT: Augmented BERT on phonemes and graphemes for neural TTS," in *Proc. Interspeech*, 2021, pp. 151–155.

[9] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Hum. Lang. Technol.*, J. Burstein, C. Doran, and T. Solorio, Eds., 2019, vol. 1, pp. 4171–4186.

[10] G. Jawahar, B. Sagot, and D. Seddah, "What does BERT learn about the structure of language?," in *Proc. 57th Conf. Assoc. Comput. Linguistics*, L. A. Papers, D. Korhonen, R. Traum, and L. Má rquez, Eds., 2019, pp. 3651–3657.

[11] I. Tenney, D. Das, and E. Pavlick, "BERT rediscovers the classical NLP pipeline," in *Proc. 57th Conf. Assoc. Comput. Linguistics*, L. A. Papers, D. Korhonen, R. Traum, and L. Má rquez, Eds., 2019, pp. 4593–4601.

[12] X. Wang, "Fundamental frequency modeling for neural-network-based statistical parametric speech synthesis," Ph.D. dissertation, Dept. Informat., SOKENDAI, Hayama, Japan, 2018.

[13] M. Aso, S. Takamichi, and H. Saruwatari, "End-to-end text-to-speech synthesis with unaligned multiple language units based on attention," in *Proc. Interspeech*, 2020, pp. 4009–4013.

[14] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 6000–6010.

[15] Y. Sagisaka and H. Sato, "Accentuation rules for Japanese word concatenation," *IEICE Trans. Inf. Syst.*, vol. 66, no. 7, pp. 849–856, 1983.

[16] M. SUZUKI et al., "Accent sandhi estimation of Tokyo dialect of Japanese using conditional random fields," *IEICE Trans. Inf. Syst.*, vol. 100, no. 4, pp. 655–661, 2017.

[17] M. Nobuaki, K. Ryo, H. Keikichi, and W. Michiko, "CRF-based statistical learning of Japanese accent sandhi for developing Japanese text-to-speech synthesis systems," in *Proc. 6th ISCA Speech Synth. Workshop*, 2007, pp. 148–153.

[18] Y. Yasuda, X. Wang, and J. Yamagishi, "Investigation of learning abilities on linguistic features in sequence-to-sequence text-to-speech synthesis," *Comput. Speech Lang.*, vol. 67, 2021, Art. no. 101183.

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

[20] R. V. Anil, T. Gupta Koren, and Y. Singer, "Memory efficient adaptive optimization," in *Proc. Adv. Neural Inf. Process. Syst. 32: Annu. Conf. Neural Inf. Process. Syst.*, H. M. H. Wallach, A. Larochelle, F. Beygelzimer d' E. B. Alché-Buc Fox, and R. Garnett, Eds., 2019, pp. 9746–9755.

[21] S. Toshinori, "Neologism dictionary based on the language resources on the web for Mecab," 2015. [Online]. Available: https://github.com/neologd/mecab-ipadic-neologd

[22] T. H. T. Sato and M. Okumura, "Operation of a word segmentation dictionary generation system called NEologd," (in Japanese) *Inf. Process. Soc. Japan, Special Int. Group Natural Lang. Process.*, 2016, pp. 1–14.

[23] T. H. T. Sato and M. Okumura, "Implementation of a word segmentation dictionary called mecab-ipadic-NEologd and study on how to use it effectively for information retrieval," (in Japanese) in *Proc. 23rd Annu. Meeting Assoc. Natural Lang. Process.*, 2017, pp. NLP2017–B6-1.

[24] R. Sonobe, S. Takamichi, and H. Saruwatari, "JSUT corpus: Free large-scale Japanese speech corpus for end-to-end speech synthesis," 2017, *arXiv:1711.00354v1*.

[25] B. Zoph and K. Knight, "Multi-source neural translation," in *Proc. NAACL-HLT*, 2016, pp. 30–34.

[26] B. McCann and R. Socher, "Learned in translation: Contextualized word vectors," in *Proc. Adv. Neural Inf. Process. Syst. 30: Annu. Conf. Neural Inf. Process. Syst.*, U. von Garnett, Ed., 2017, pp. 6294–6305.

[27] J.-X. Zhang, Z.-H. Ling, and L.-R. Dai, "Forward attention in sequence-to-sequence acoustic modeling for speech synthesis," in *Proc. Int. Conf. Acoust., Speech Signal Process.*, 2018, pp. 4789–4793.

[28] T. Hayashi, S. Watanabe, T. Toda, K. Takeda, S. Toshniwal, and K. Livescu, "Pretrained text embeddings for enhanced text-to-speech synthesis," in *Proc. 20th Annu. Conf. Int. Speech Commun. Assoc.*, G. Kubin and Z. Kacic, Eds., 2019, pp. 4430–4434.

[29] H. Tachibana, K. Uenoyama, and S. Aihara, "Efficiently trainable text-to-speech system based on deep convolutional networks with guided attention," in *Proc. IEEE Int. Conf. Acoust.*, *Speech Signal Process.*, 2018, pp. 4784–4788.

[30] Y. Wang et al., "Tacotron: Toward end-to-end speech synthesis," in *Proc. Interspeech*, 2017, pp. 4006–4010.

[31] R. Yamamoto, E. Song, and J. Kim, "Parallel waveGAN: A fast waveform generation model based on generative adversarial networks with multiresolution spectrogram," in *Proc. IEEE Int. Conf. Acoust.*, *Speech Signal Process.*, 2020, pp. 6199–6203.

[32] K. MAEKAWA, "X-JTOBI : An extended J-TOBI for spontaneous speech," in *Proc. 7th ICSLP*, 2002, pp. 1545–1548.

[33] S. Watanabe et al., "ESPnet: End-to-end speech processing toolkit," in *Proc.19th Annu. Conf. Int. Speech Commun. Assoc.*, 2018, pp. 2207–2211.

[34] H. B. Mann and D. R. Whitney, "On a test of whether one of two random variables is stochastically larger than the other," *Ann. Math. Statist.*, vol. 18, no. 1, pp. 50–60, 1947.

[35] O. S. Watts, "Unsupervised learning for text-to-speech synthesis," Ph.D. dissertation, Univ. Edinburgh, Edinburgh, Scotland, 2013.

[36] H. Lu, S. King, and O. Watts, "Combining a vector space representation of linguistic context with a deep neural network for text-to-speech synthesis," in *Proc. 8th ISCA Workshop Speech Synth.*, 2013, pp. 261–265.

[37] J. Pan et al., "A unified sequence-to-sequence front-end model for mandarin text-to-speech synthesis," in *Proc. IEEE Int. Conf. Acoust.*, 2020, pp. 6689–6693.

[38] J. Li, Z. Wu, R. Li, P. Zhi, S. Yang, and H. Meng, "Knowledge-based linguistic encoding for end-to-end mandarin text-to-speech synthesis," in *Proc. Interspeech*, 2019, pp. 4494–4498.

[39] T. Mikolov, M. Karafiát, L. Burget, J. Cernocký, and S. Khudanpur, "Recurrent neural network based language model," in *Proc.*, 11th *Ann. Conf. Int. Speech Commun. Assoc.*, 2010, pp. 1045–1048.

[40] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *Proc. 1st Int. Conf. Learn. Representations*, 2013, pp. 2–4.

[41] P. Wang, Y. Qian, F. K. Soong, L. He, and H. Zhao, "Word embedding for recurrent neural network based TTS synthesis," in *Proc. IEEE Int. Conf. Acoust.*, 2015, pp. 4879–4883.

[42] W. Fang, Y. Chung, and J. R. Glass, "Toward transfer learning for end-to-end speech synthesis from deep pre-trained language models," 2019, *arXiv:1906.07307v1*.

[43] H. Ming, L. He, H. Guo, and F. K. Soong, "Feature reinforcement with word embedding and parsing information in neural TTS," 2019, *arXiv:1901.00707v2*.

[44] Y. Xiao, L. He, H. Ming, and F. K. Soong, "Improving prosody with linguistic and bert derived features in multi-speaker based mandarin chinese neural TTS," in *Proc. IEEE Int. Conf. Acoust.*, 2020, pp. 6704–6708.

[45] G. Xu, W. Song, Z. Zhang, C. Zhang, X. He, and B. Zhou, "Improving prosody modelling with cross-utterance bert embeddings for end-to-end speech synthesis," in *Proc. IEEE Int. Conf. Acoust.*, 2021, pp. 6079–6083.

[46] W. Nakata et al., "Audiobook speech synthesis conditioned by cross-sentence context-aware word embeddings," in *Proc. 11th ISCA Speech Synth. Workshop* (SSW 11), 2021, pp. 211–215.

[47] S. Tyagi, M. Nicolis, J. Rohnke, T. Drugman, and J. Lorenzo-Trueba, "Dynamic prosody generation for speech synthesis using linguistics-driven acoustic embedding selection," in *Proc. 21st Annu. Conf. Int. Speech Commun. Assoc.*, 2020, pp. 4407–4411.

[48] T. Kenter, M. Sharma, and R. Clark, "Improving the prosody of RNN-based english text-to-speech synthesis by incorporating a BERT model," in *Proc. 21st Annu. Conf. Int. Speech Commun. Assoc., Virtual Event*, 2020, pp. 4412–4416.

[49] S. Karlapati, "Prosodic representation learning and contextual sampling for neural text-to-speech," in *Proc. IEEE Int. Conf. Acoust.*, 2021, pp. 6573–6577.

[50] Y. Chung, Y. Wang, W. Hsu, Y. Zhang, and R. J. Skerry-Ryan, "Semi-supervised training for improving data efficiency in end-to-end speech synthesis," in *Proc. IEEE Int. Conf. Acoust.*, 2019, pp. 6940–6944.

[51] M. E. Peters, S. Ruder, and N. A. Smith, "To tune or not to tune? Adapting pretrained representations to diverse tasks," in *Proc. 4th Workshop Representation Learn.* I. S. Augenstein et al., Eds., 2019, pp. 7–14.

[52] Q. Chen, W. Wang, and Q. Zhang, "Pre-training for spoken language understanding with joint textual and phonetic representation learning," in *Proc. Interspeech*, 2021, pp. 1244–1248.

[53] L. Chen, Y. Deng, X. Wang, F. K. Soong, and L. He, "Speech bert embedding for improving prosody in neural TTS," in *Proc. IEEE Int. Conf. Acoust.*, *Speech Signal Process.*, 2021, pp. 6563–6567.

Yusuke Yasuda (Member, IEEE) received the B.S. and M.S. degrees from Waseda University, Tokyo, Japan, in 2012 and 2014, respectively, and the Ph.D. degree from the Graduate university for Advanced Studies, SOKENDAI, Hayama, Japan, in 2021. From 2018 to 2021, he was a Research Assistant with the National Institute of Informatics, Tokyo, Japan. Since 2021, he has been an Assistant Project Professor with the Information Technology Center, Nagoya University, Nagoya, Japan. His research interests include statistical machine learning and speech synthesis. He was the recipient of the 20th Best Student Paper Award from the Acoustical Society of Japan.

Tomoki Toda (Senior Member, IEEE) received the B.E. degree from Nagoya University, Tokyo, Japan, in 1999 and the M.E. and D.E. degrees from Nara Institute of Science and Technology (NAIST), Ikoma, Japan, in 2001 and 2003, respectively. From 2003 to 2005, he was a Research Fellow of the Japan Society for the Promotion of Science. Then, he was an Assistant Professor from 2005 to 2011 and an Associate Professor from 2011 to 2015 with NAIST. Since 2015, he has been a Professor with the Information Technology Center, Nagoya University, Nagoya, Japan. His research focuses on statistical approaches to sound media information processing. He was the recipient of more than 15 article/achievement awards, including the IEEE SPS 2009 Young Author Best Paper Award and the 2013 EURASIP-ISCA Best Paper Award (Speech Communication Journal).