INTERSPEECH 2021
30 August – 3 September, 2021, Brno, Czechia

Phrase break prediction with bidirectional encoder representations in Japanese text-to-speech synthesis

Kosuke Futamata, Byeongseon Park, Ryuichi Yamamoto, Kentaro Tachibana

LINE Corp., Tokyo, Japan

{kosuke.futamata, park.byeongseon, ryuichi.yamamoto, kentaro.tachibana}@linecorp.com

Abstract

We propose a novel phrase break prediction method that combines implicit features extracted from a pre-trained large language model, a.k.a BERT, and explicit features extracted from BiLSTM with linguistic features. In conventional BiLSTM-based methods, word representations and/or sentence representations are used as independent components. The proposed method takes account of both representations to extract the latent semantics, which cannot be captured by previous methods. The objective evaluation results show that the proposed method obtains an absolute improvement of 3.2 points for the F1 score compared with BiLSTM-based conventional methods using linguistic features. Moreover, the perceptual listening test results verify that a TTS system that applied our proposed method achieved a mean opinion score of 4.39 in prosody naturalness, which is highly competitive with the score of 4.37 for synthesized speech with ground-truth phrase breaks.

Index Terms: text-to-speech front-end, phrase break prediction, speech synthesis, BERT, pre-trained text representations

1. Introduction

Deep learning-based end-to-end approaches have achieved great success in text-to-speech (TTS) synthesis. In particular, TTS systems based on sequence-to-sequence models (e.g., Tacotron [1, 2]) enable the models to directly map character sequences to acoustic features, thereby eliminating the need for a complicated text processing front-end. However, front-end modules, such as language-dependent text normalization and grapheme-to-phoneme (G2P) conversion, are still beneficial as pre-processing steps, and can be crucial for practical applications [3, 4]. Furthermore, in addition to text normalization and G2P conversion, prosodic features, including an accent phrase and phrase break, also play an important role in pitch-accent languages like Japanese [5, 6].

The front-end of a TTS system depends on the language. For Japanese, it includes a variety of modules, such as text normalization, accent estimation [7], G2P conversion [8, 9], and phrase break prediction [10, 11]. In this study, we focus on Japanese phrase break prediction, which is one of the essential tasks for improving the naturalness of the prosody.

Previous studies can be divided into two categories. One is traditional statistical methods [10, 12, 13] using manually designed linguistic features. The other category includes deep learning-based sequential models, such as Reccurent Neural Network (RNN) [14–16], and Long-Short Term Memory (LSTM) [16, 17]. These methods achieve much better performance than traditional statistical methods. However, the main problem with these methods is that they usually require a large amount of labeled data to achieve good performance. Therefore, conventional methods have made use of unsupervised pre-trained representations, including word representations and/or sentence representations [16–19] to improve performance, even with small data. Although these unsupervised representations are helpful, their performance is potentially limited because word representations and sentence representations are typically extracted by separate modules without modeling complex semantic structures.

To address these issues of previous methods, inspired by the great success of Bidirectional Encoder Representations from Transformers (BERT) [20], which is an unsupervised pre-trained large language model applied in many NLP tasks, and the success of BERT in Chinese phrase break prediction [21], we propose a novel method that uses both labeled and unlabeled data for phrase break prediction. The proposed method uses both explicit features from LSTM using various linguistic features and implicit features from BERT to model the relations between features and phrase break information. Because BERT can learn latent word and sentence representations with semantic meanings simultaneously, it provides a better representation than simple word or sentence embedding representations. Additionally, because the model can use explicit features extracted from LSTM using various linguistic features as input, it makes it possible for the model to improve performance, even for a limited amount of training data.

We compared the proposed method in both an objective evaluation based on the F1 score and subjective evaluations based on the mean opinion score (MOS) test and AB preference test. The results of the objective evaluation showed that the proposed method helped the model to achieve a significant improvement of 3.2 points for the F1 score compared with a BiLSTM-based conventional method using linguistic features. Moreover, the results of the subjective evaluations verified that the TTS system that applied our proposed method achieved an MOS of 4.39 in terms of prosody naturalness, which is highly competitive with the score of 4.37 for synthesized speech with ground-truth phrase breaks. Additionally, there were significant differences between the proposed method and the conventional methods for the AB preference test.

2. Phrase break prediction in TTS front-end

A phrase break is defined as a phonetic pause inserted between phrases, and it occurs because of breathing and/or an intonational reset. In a text, phrase breaks are usually represented as a type of punctuation. When the TTS system observes a comma in English text, it is generally accepted that a phrase break should be inserted, as the author intends the words before and after the punctuation to be separated. However, this rule-based approach usually does not work well. It has been reported that the rule-based approach results in about half of the phrase breaks being inserted correctly, and a few breaks are inserted at incorrect positions [22]. These errors come from the

http://dx.doi.org/10.21437/Interspeech.2021-252

fact that human speakers usually insert phrase breaks without punctuation, for example, for breathing, expression, and accent change.

The main goal of phrase break prediction is to predict the positions of superficial and latent phrase breaks from the input text. This task can be formulated as sequential labeling, which labels a phrase break or non-phrase break for each token after applying some text pre-processing modules, such as text normalization and tokenization; that is, given a source sequence $X = \{x_1, x_2, ..., x_t\}$ and label sequence $Y = \{y_1, y_2, ..., y_t\}$ corresponding to a source sequence X, the goal is to predict a label sequence Y from a token sequence X, and the i-th element in label sequence Y is defined as a binary label with a non-phrase break (y_i = 0) or phrase break (y_i = 1). Given the model parameter θ and source sequence X, we aim to calculate parameters $\hat{\theta}$ by maximizing the log-likelihood of the given label sequence Y as follows:

$$\hat{\theta} = \arg\max_{\theta} \sum_{i=1}^{t} \log p(Y^{(i)}|X^{(i)}; \theta) \quad (1)$$

The deep learning-based model architecture has been successfully applied to phrase break prediction. Applications of bidirectional LSTM (BiLSTM) have been studied as sequence labeling tasks [16–19]. These methods use part-of-speech (POS) tags, dependency tree features, pre-trained word embedding, and character embedding in addition to the input sequences to improve performance. Using BiLSTM enables the model to learn long-term dependencies from both the left and right direction, and add a variety of linguistic features as the input enriches contextual information in long-form dependencies. In previous studies, not only a source sequence but also these linguistic features were used to predict phrase breaks.

3. Proposed method using bidirectional encoder representations

The main problem with the TTS front-end is that it costs a great deal to prepare a large labeled corpus. Therefore, in conventional approaches, unsupervised pre-training methods, such as word embedding and sentence representations using a BiLSTM-based language model, are used to improve performance. However, these representations are weak context representations; pre-trained word embedding contains semantic information for only a single word without sequential dependencies, and the BiLSTM language model lacks sufficient contextual information for a much longer sequence, although it learns long-term dependencies more easily than vanilla RNN. To solve these problems, BERT [20] model, which is a recently proposed NLP pre-training method, is widely used to capture the long-term context dependency in sequences.

3.1. BERT architecture

BERT is a recently proposed language model composed of bidirectional hierarchical Transformer [23] blocks. The model can be used as an encoder that takes a series of subword tokens as input and generates a word embedding for each token. The multi-head self-attention mechanism in Transformer blocks enables the model to capture word dependencies for both the left and right side context without any restriction on the position of words in a sentence. BERT is fine-tuned for each task after two unsupervised pre-training tasks called masked language modeling (MLM), as well as next sentence prediction (NSP). Benefit from this pre-training, the model is assumed to be capable of capturing rich contextual and semantic information of Japanese language, thereby facilitating the downstream NLP tasks (i.e., Phrase break prediction in this work).

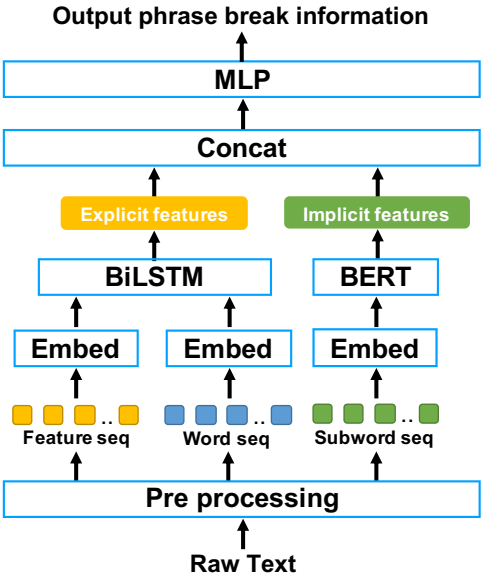

Figure 1: *Architecture of the proposed method using BiLSTM with BERT for phrase break prediction*

3.2. Application to phrase break prediction

The proposed method predicts phrase break positions using implicit features extracted from BERT and explicit features extracted from the conventional method, BiLSTM. Figure 1 shows the architecture of our proposed method.

The input Japanese text is firstly pre-processed by text normalization, tokenization, POS tagging, and dependency parsing modules. Explicit and implicit features are then extracted by BERT and BiLSTM encoder. On the BiLSTM side, the word sequence, POS tags, and dependency tree features corresponding to the sequence are fed into the model, and then explicit features are generated from the last layer. Word embedding in BiLSTM is pre-trained on Japanese Wikipedia. Conversely, on the BERT side, a subword sequence is fed into the model, and then implicit features are generated from a weighted average of all the layers, where the weights are the trainable parameters. We use a weighted average of all the layers instead of using only the last layer of BERT, which is generally used for NLP downstream tasks, because different BERT layers have different types of information [24]. The lower layers have the most linear word order information, the middle layers have the most syntactic information, and the final layer has the most task-specific information. Preliminary experiments showed that using a weighted average of all the layers contributed much more to performance than using only the last layer.

Explicit features from BiLSTM and implicit features from BERT are finally concatenated, and then the method determines whether a phrase break should be labeled after each token. We expect that implicit features extracted from BERT will learn the rich syntactic and semantic information of each word in a given context, which cannot be captured by only adding the BiLSTM

Table 1: *Performance of different systems on Japanese for the phrase break prediction task*

System	Model	Input features	True positives	False negatives	False positives	F1	Precision	Recall
1	BiLSTM (Tokens)	Tokens	653	114	53	88.7	92.5	85.1
2	BiLSTM (Features)	Tokens+POS+DEP+W2V	676	91	56	90.2	92.3	88.1
3	BERT	Tokens	688	79	39	92.1	**94.6**	89.7
4	BiLSTM (Tokens) + BERT	Tokens	690	77	42	92.1	94.3	90.0
5	**BiLSTM (Features) + BERT**	Tokens+POS+DEP+W2V	709	58	43	**93.4**	94.3	**92.4**

Table 2: *Performance of different systems on Japanese for phrase break prediction with and without punctuation*

System	With punctuation						Without punctuation					
	True positives	False negatives	False positives	F1	Precision	Recall	True positives	False negatives	False positives	F1	Precision	Recall
1	357	0	1	99.9	99.7	100	296	114	52	78.1	85.1	72.2
2	357	0	1	99.9	99.7	100	319	91	55	81.4	85.3	77.8
3	355	2	0	99.7	100	99.4	333	77	39	85.2	**89.5**	81.2
4	356	1	1	99.7	99.7	99.7	334	76	41	85.1	89.1	81.5
5	357	0	1	99.9	99.7	100	352	58	42	**87.6**	89.3	**85.9**

Table 3: *MOS results with 95% confidence intervals*

System	Model	MOS
1	Reference (Natural)	4.70 ± 0.05
2	Reference (TTS)	4.37 ± 0.05
3	Rule-based	3.62 ± 0.07
4	BiLSTM (Tokens)	3.80 ± 0.07
5	BiLSTM (Features)	4.05 ± 0.06
6	BERT	4.26 ± 0.05
7	**BiLSTM (Features) + BERT**	$\mathbf{4.39} \pm 0.05$

layer. Conversely, we expect that explicit features extracted from BiLSTM should capture the long-term dependencies in a source sequence directly, thereby helping the implicit features to be much more rich representations.

4. Objective evaluation

4.1. Experimental setting

We compared the performance of the proposed method with different systems based on either BiLSTM or BERT using an objective evaluation. For the dataset, we collected 99,907 utterances, where each utterance was transcribed and had silence between words, and word transitions with a silence of more than 200 ms were marked as phrase breaks. We split the dataset into three subsets of 98,807, 500, and 500 sentences for training, validation, and test, respectively.

In the objective evaluation, we compared the following five methods. (1) **BiLSTM (Tokens)**: conventional BiLSTM-based method using only a source sequence [16]. (2) **BiLSTM (Features)**: (1) BiLSTM (Tokens)-based method that adds designed linguistic features [18], including POS tags [25], dependency tree features (DEP) [26], and pre-trained word embedding. (3) **BERT**: method that uses BERT model. (4) **BiLSTM (Tokens) + BERT**: method that combines (1) BiLSTM (Tokens) and (3) BERT. (5) **BiLSTM (Features) + BERT**: proposed method that combines (2) BiLSTM (Features) and (3) BERT. We set up baselines as conventional approaches; (1) BiLSTM (Tokens) and BiLSTM (Features).

For all experiments, we used a two-layer BiLSTM (with 512 dimensions). We set the embedding sizes of linguistic features to 32. The model was trained for 20 epochs with Adam optimizer [27] with a minibatch size of 64 utterances. The learning rate was held constant at 1e-5. The loss function that we used to train the model was cross-entropy loss, and we evaluated the model using the F1 score. We stopped training when the F1 score did not increase for 10 epochs. For BiLSTM (Features) and BiLSTM (Features) + BERT, we used word embedding vectors pre-trained on Japanese Wikipedia articles. For BERT-based models, we used a BERT-Base model pre-trained on Japanese Wikipedia articles and released on Github [28].

4.2. Results and analysis

The experimental results of different systems are presented in Table 1 and Table 2. Table 1 shows all results regardless of whether a phrase break was provided by punctuation in the utterance, and Table 2 shows separate results when a phrase break was provided with and without punctuation.

The experimental results presented in Table 1 show that the proposed method improved the F1 score compared with the conventional methods. Comparing BERT-based models with the conventional methods, BiLSTM (Tokens) and BiLSTM (Features), the BERT-based models improved the F1 score significantly, This implies that the proposed method achieved high performance, even for a limited amount of training data. The proposed method achieved absolute improvements of 4.7 points and 3.2 points compared with BiLSTM (Tokens) and BiLSTM (Features), respectively, which indicates that implicit features extracted from BERT are much more effective than explicit features only used in the conventional methods. Furthermore, the F1 score of the proposed method was much higher than that of BiLSTM (Tokens) + BERT, although the performances of BERT and BiLSMT (Tokens) + BERT were the same. This implies that explicit features extracted from BiLSTM using various linguistic features were used effectively for predicting phrase breaks.

5. Subjective Evaluation

5.1. Experimental setting

5.1.1. TTS setup

To verify the effectiveness of the proposed phrase break prediction method in TTS scenarios, we combined the proposed method as part of the text processing front-end of a FastSpeech2

Table 4: *AB preference results for the one-tailed binomial test, n.s.: no difference, ***: significant differences with* $p \leq .001$

Model A	Model B	Preference A	Preference B	Neutral	binomial test
Rule-based	BiLSTM (Tokens)	14.4%	**38.5%**	47.1%	***
BiLSTM (Tokens)	BiLSTM (Features)	19.8%	**32.1%**	48.1%	***
BiLSTM (Features)	BERT	15.9%	**37.8%**	46.3%	***
BERT	BiLSTM (Features) + BERT	16.6%	**27.5%**	56.1%	***
BERT	Reference (TTS)	17.9%	**31.0%**	51.1%	***
BiLSTM (Features) + BERT	Reference (TTS)	17.4%	**17.9%**	64.7%	n.s.

based TTS system [29]. The TTS system consisted of two models: (1) a feed-forward Transformer-based acoustic model that predicts acoustic features from a phoneme sequence with additional phrase break information, and (2) a parallel WaveGAN vocoder that generates speech waveforms from acoustic features [30]. The detailed model structure and training conditions of these two models were the same as those in [31]. By inputting the predicted phrase breaks along with the input phoneme sequence, the TTS system generated target speech waveforms.

As a database, we used the same corpus as that used in the phrase break prediction experiments. The speech corpus was recorded by a single Japanese professional speaker. The speech signals were sampled at 24 kHz, and each sample was quantized by 16 bits. The total amount of the training data size was 118.9 hours. Note that we used the same test set as that used in phrase break prediction, which was not included in the training set, for evaluations.

5.1.2. Subjective evaluation setup

We performed an MOS test and AB preference test for the subjective evaluations. In the MOS test, 29 native Japanese speakers were asked to make quality judgments about the synthesized speech samples using the following five possible responses: 1 = Bad; 2 = Poor; 3 = Fair; 4 = Good; 5 = Excellent. In the AB preference test, each test case consisted of two audio samples, A and B, where A and B correspond to the synthesized speech samples of different systems. The same subjects of the MOS test were asked to choose the audio sample A or B that was more natural than the other, or state whether A and B were the same in terms of the naturalness of the prosody.

For the MOS test, we compared the following seven systems: (1) **Reference (Natural)**: recorded speech in the test set; (2) **Reference (TTS)**: synthesized speech from the test set (the position of phrase breaks is the same with recorded speech); (3) **Rule-based**: a rule-based method that inserts phrase breaks only after punctuation, and some methods used in the objective evaluations; (4) **BiLSTM (Tokens)**, (5) **BiLSTM (Features)**, (6) **BERT**, and (7) **BiLSTM (Features) + BERT**. For the AB preference test, we compared the following six system pairs; (3) Rule-based and (4) BiLSTM (Tokens); (4) BiLSTM (Tokens) and (5) BiLSTM (Features); (5) BiLSTM (Features) and (6) BERT; (6) BERT and (7) BiLSTM (Features) + BERT; (6) BERT and (2) Reference (TTS); and (7) BiLSTM (Features) + BERT and (2) Reference (TTS). Note that synthesized speech can manually be adjusted by specifying the position of phrase breaks. Audio samples for each system are available on the website[1].

For the MOS test, 30 utterances were randomly selected from the test set and then synthesized using the above seven models; in total, 210 utterances were used. For the AB preference test, the same utterances used for the MOS test were selected from the test set and then synthesized using the above six models; in total, 360 utterances were used.

[1] `https://matasuke.github.io/demos/pbp_bert`

5.2. Results and analysis

Table 3 shows the MOS test results for the different systems. The experimental results of the MOS test showed that the proposed method achieved almost the same quality as Reference (TTS), which synthesized test data, and indicated a significant improvement in the naturalness of the prosody compared with conventional methods. Comparing BiLSTM (Features) + BERT with the conventional methods, BiLSTM (Tokens) and BiLSTM (Features), the proposed method achieved 4.39 MOS points and improved by 0.59 points and 0.34 points, respectively. Additionally, the proposed method improved by 0.13 MOS points compared with BERT. This indicates that using both explicit features from BiLSTM and implicit features from BERT was effective.

Table 4 shows the AB preference test results with respect to different systems: n.s. denotes no significant difference, and *** denotes statistically significant difference at the 1% level as a result of a one-tailed binomial test. The experimental results of the AB preference test showed significant differences at the 1% level for five pairs, except for the results between BiLSTM (Features) + BERT and Reference (TTS). For the pair between BiLSTM (Features) + BERT and Reference (TTS), the preferences for each system were competitive, and there was no significant difference between them. Hence, we assume that the proposed method achieved almost the same quality as the synthesized speech samples using the reference text.

6. Conclusions

In this paper, inspired by the great success of BERT in many NLP tasks, we presented a novel method using both labeled data and unlabeled data for phrase break prediction in a Japanese TTS front-end. The objective evaluation results showed that the BiLSTM and BERT-based proposed method significantly improved the performance of phrase break prediction compared with conventional methods and the model based only on BERT. Moreover, subjective evaluation results verified that the proposed method achieved almost the same quality as the synthesized speech samples using reference text in terms of the naturalness of the prosody. In the future, we will aim to further improve the quality of synthesized speech samples in terms of the naturalness of the prosody by modeling the duration of phrase breaks.

7. Acknowledgements

This work was supported by Clova Voice, NAVER Corp., Seongnam, Korea.

8. References

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

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

[3] N. Li, S. Liu, Y. Liu, S. Zhao, M. Liu, and M. T. Zhou, "Neural speech synthesis with Transformer network," in *Proc. AAAI*, 2019, pp. 6706–6713.

[4] K. Kastner, J. F. Santos, Y. Bengio, and A. Courville, "Representation mixing for tts synthesis," in *Proc. ICASSP*. IEEE, 2019, pp. 5906–5910.

[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. SSW*, 2019, pp. 166–171.

[6] 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, 2021.

[7] N. Minematsu, S. Kobayashi, S. Shimizu, and K. Hirose, "Improved prediction of japanese word accent sandhi using crf," in *Proc. INTERSPEECH*, vol. 10, no. 38,900, 2012, pp. 114–783.

[8] K. Rao, F. Peng, H. Sak, and F. Beaufays, "Grapheme-to-phoneme conversion using long short-term memory recurrent neural networks," in *Proc. ICASSP*, 2015, pp. 4225–4229.

[9] Y. Sevinj, N. Géza, and B. G.-T, "Transformer based grapheme-to-phoneme conversion," in *Proc. INTERSPEECH*, 2019, pp. 2095–2099.

[10] H. Muto, Y. Ijima, N. Miyazaki, H. Mizunoand, and S. Sakauchi, "Analysis and evaluation of factors relating pause location for natural text-to-speech synthesis," *Transactions of Information Processing Society of Japan*, pp. 993–1002, 2015.

[11] H. Fujisaki, S. Ohno, and S. Yamada, "Analysis of occurrence of pauses and their durations in japanese text reading," in *Proc. ICSLP*, vol. 4, 1998, pp. 1387–1390.

[12] Y. Qian, Z. Wu, X. Ma, and F. Soong, "Automatic prosody prediction and detection with conditional random field (crf) models," in *Proc. ISCSLP*, 2010, pp. 135–138.

[13] D.-S. Na and M.-J. Bae, "A variable break prediction method using cart in a japanese text-to-speech system," *IEICE Transactions on Information and Systems*, vol. E92.D, no. 2, pp. 349–352, 2009.

[14] S. Pascual and A. Bonafonte, "Prosodic break prediction with rnns," in *Proc. IberSPEECH*, 2016, pp. 64–72.

[15] M. Fishel and M. Mihkla, "Modelling the temporal structure of newsreaders' speech on neural networks for estonian text-to-speech synthesis," in *Proc. SPECOM*, 2006, pp. 303–306.

[16] A. Vadapalli and S. V. Gangashetty, "An investigation of recurrent neural network architectures using word embeddings for phrase break prediction," in *Proc. INTERSPEECH*, 2016, pp. 2308–2312.

[17] Y. Zheng, J. Tao, Z. Wen, and Y. Li, "Blstm-crf based end-to-end prosodic boundary prediction with context sensitive embeddings in a text-to-speech front-end," in *Proc. INTERSPEECH*, 2018, pp. 47–51.

[18] V. Klimkov, A. Nadolski, A. Moinet, B. Putrycz, R. Barra-Chicote, T. Merritt, and T. Drugman, "Phrase break prediction for long-form reading tts: Exploiting text structure information," in *Proc. INTERSPEECH*, 2017, pp. 1064–1068.

[19] O. Watts, S. Gangireddy, J. Yamagishi, S. King, S. Renals, A. Stan, and M. Giurgiu, "Neural net word representations for phrase-break prediction without a part of speech tagger," *Proc. ICASSP*, pp. 2599–2603, 2014.

[20] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. NAACL-HLT*, 2019, pp. 4171–4186.

[21] 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, ICASSP*, 2020, pp. 6704–6708.

[22] P. Taylor, *Text-to-Speech Synthesis*. New York: Cambridge Univerty press, 2009.

[23] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. NIPS*, 2017, pp. 5996–6008.

[24] R. Anna, K. Olga, and R. Anna, "A primer in BERTology: What we know about how BERT works," *Transactions of the Association for Computational Linguistics*, vol. 8, pp. 842–866, 2020.

[25] K. Takaoka, S. Hisamoto, N. Kawahara, M. Sakamoto, Y. Uchida, and Y. Matsumoto, "Sudachi: a japanese tokenizer for business," in *Proc. LREC*, 2018.

[26] https://github.com/megagonlabs/ginza.

[27] D. Kingma and J. Ba, "Adam: A method for stochastic optimization." in *Proc. ICLR*, 2015.

[28] https://github.com/cl-tohoku/bert-japanese.

[29] R. Yi, H. Chenxu, Q. Tao, Z. Sheng, Z. Zhou, and L. Tie-Yan, "FastSpeech 2: Fast and high-quality end-to-end text-to-speech," in *Proc. ICLR (in press)*, 2021.

[30] R. Yamamoto, E. Song, and J.-M. Kim, "Parallel WaveGAN: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram," in *Proc. ICASSP*, 2020, pp. 6199–6203.

[31] R. Yamamoto, E. Song, M.-J. Hwang, and J.-M. Kim, "Parallel waveform synthesis based on generative adversarial networks with voicing-aware conditional discriminators," in *Proc. ICASSP (in press)*, 2021.