INTERSPEECH 2020
October 25–29, 2020, Shanghai, China

End-to-end text-to-speech synthesis with unaligned multiple language units based on attention

Masashi Aso, Shinnosuke Takamichi, and Hiroshi Saruwatari

Graduate School of Information Science and Technology, The University of Tokyo, Japan
716asmssh@gmail.com, {shinnosuke_takamichi,hiroshi_saruwatari}@ipc.i.u-tokyo.ac.jp

Abstract

This paper presents the use of unaligned multiple language units for end-to-end text-to-speech (TTS). End-to-end TTS is a promising technology in that it does not require intermediate representation such as prosodic contexts. However, it causes mispronunciation and unnatural prosody. To alleviate this problem, previous methods have used multiple language units, e.g., phonemes and characters, but required the units to be hard-aligned. In this paper, we propose a multi-input attention structure that simultaneously accepts multiple language units without alignments among them. We consider using not only traditional phonemes and characters but also subwords tokenized in a language-independent manner. We also propose a progressive training strategy to deal with the unaligned multiple language units. The experimental results demonstrated that our model and training strategy improve speech quality.

Index Terms: End-to-end, Text-to-speech, Subword, Progressive training, Transformer

1. Introduction

Text-to-speech (TTS) [1, 2, 3] converts any text to corresponding speech. Recently, end-to-end models for TTS have gained attention. End-to-end models [4, 5, 6, 7] have various advantages, compared with previous cascaded models (a.k.a., statistical parametric speech synthesis) [8, 9]. For example, end-to-end models require less language knowledge than statistical parametric speech synthesis. These models can convert text to audio directly without such intermediate features. One successful end-to-end model is Tacotron2 [5], which can synthesize high-fidelity speech from character sequences. However, it often causes mispronunciations [10, 11] and predicts prosody poorly.

The use of multiple language units other than characters is expected to solve this problem. For example, the usage of phonemes will decrease mispronunciation [6, 12] because phonemes are more related to speech than characters. There are two methods to use multiple language units; random usage and simultaneous usage. One example of the random usage is DeepVoice3 [6], which is trained with randomly selected characters and phonemes. The developers of DeepVoice3 reported that the usage of both characters and phonemes reduces the counts of mispronunciation. Moreover, it enables us to correct the pronunciation manually by replacing mispronounced characters with phonemes. The representation mixing model [12] also use randomly character and phoneme, and it shows that a model with a mixed representation of characters and phonemes surpasses one with only characters in synthesized speech quality. A number of examples of simultaneous usage include Chinese end-to-end TTS [13, 14] because prosody is linguistically essential for Chinese and it is difficult to predict the prosody only with phonemes. Lu et al. [13] used a part of full context labels in addition to phonemes to better predict prosody. Zhu [14] reduced misprediction of sandhi tones by incorporating the BERT [15] embedding of characters. However, these methods with simultaneous usage requires alignment among the language units. For example, Chinese TTS with BERT embedding [14] requires concatenating character and phoneme embeddings, which requires alignments between them. The alignments among language units, unlike Chinese, are not obvious in many language and the application of methods is limited.

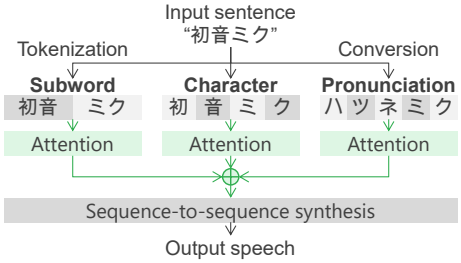

Figure 1: *Overview of our method. To condition the TTS model, our method uses multiple language units, i.e., pronunciation symbols, characters, and subwords, without their alignment.*

Thus, we propose end-to-end TTS that uses multiple language units as input without alignments among them. Fig. 1 shows an overview. Our model integrates unit-wise attention scores to condition the speech synthesis. Since the unit-wise attentions individually contribute to the speech synthesis, we do not require alignments among the language units. In addition to phonemes and characters used in previous methods, we consider the use of subwords as input, which are tokenized from a sentence without language knowledge. Subwords [16, 17] were originally proposed to deal with the traditional open-vocabulary problem [18]. Subword tokenization breaks up rare words into subword units (e.g., “language” into “lang” and “uage”). A subword unit is longer than a phoneme or a character, and the prosody is a phonological property of a longer unit than a phoneme or a character. Thus, the use of subwords is thought to be more helpful to predict prosody than that of phonemes or characters. To enhance the synthetic speech quality of our methods, we further propose a progressive training strategy for the multiple language units. Inspired by the progressive growth of generative adversarial networks [19], the weight of each language unit is scheduled so that the model first learns segmental features followed by suprasegmental features. Experimental results demonstrated that 1) our method significantly improves synthetic speech quality, and 2) our progressive training further improves the quality.

2. Conventional TTS with a single language unit

In this section, we describe the conventional DeepVoice3-based architecture [6, 20] with a single language unit. As shown in

http://dx.doi.org/10.21437/Interspeech.2020-2347

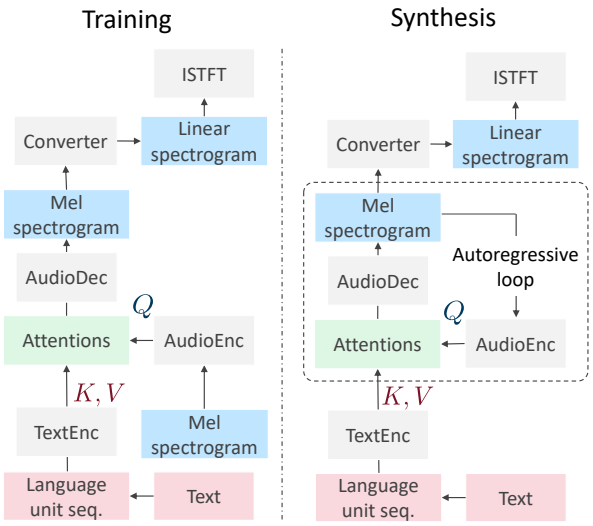

Figure 2: *End-to-end TTS used in this paper. Gray: function, Green: attention structure, Blue: speech features, Red: language features. The difference between the proposed and conventional methods lies in the "Attentions" block. In accordance with the implementation [21], we use two attentions within the "Attentions" block. "ISTFT" refers to inverse short-time Fourier transform.*

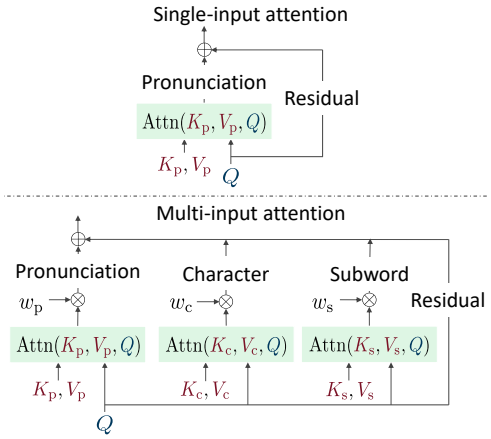

Figure 3: *Example of conventional(above) and proposed (below) attention architectures. In this example, we use pronunciation symbols for the conventional architecture, and pronunciation symbols, characters, and subwords for the proposed architecture.*

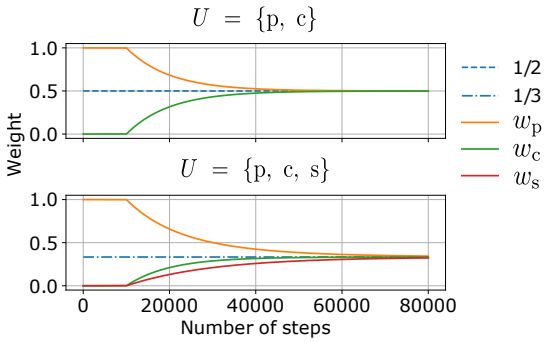

Figure 4: *The schedules of weights w_* in our methods. Subscripts "p," "c," "s" indicate pronunciation, character, and subword, respectively. Top and bottom figures are schedules for two and three language units, respectively. The x-axis ends at 80,000 steps but training will continue until 450,000 steps.*

Fig. 2, the conventional models consists of three parts; the encoder, decoder, and attention.

The encoder ("TextEnc" and "AudioEnc" in Fig. 2) outputs key K, value V, and query Q, to be used in the attention. After converting text into a sequence of pronunciation symbols, "TextEnc" transforms the sequence into K and V, and "AudioEnc" converts a sequence of speech features into Q.

The decoder ("AudioDec", "Converter", "ISTFT" in Fig. 2) synthesizes speech from the outputs of the attention. "AudioDec" outputs a mel spectrogram from the outputs of attention and predicts stop tokens, which indicates whether speech has ended. "Converter" [6, 20] predicts a high-resolution linear spectrogram from the intermediate features.

The attention architecture ("Attentions" in Fig. 2) connects the encoder and decoder. The scaled dot-product attention $Attn(\cdots)$ [22] in "Attentions" outputs attention scores on the basis of the encoder results Q, K, and V, as follows.

$$\mathrm{Attn}(K, V, Q) = \mathrm{Linear}(\mathrm{softmax}(\frac{QK^\top}{\sqrt{d_k}})V), \qquad (1)$$

where "Linear" denotes linear projection, d_k denotes the dimensionality of K, and "softmax" denotes the softmax function. These attention scores $\mathrm{Attn}(K, V, Q)$ are used as decoder inputs. We calculate the attention scores with pronunciation as input. After the calculation, we add Q as well as ResNet [23], and pass the result to the decoder.

$$Q + \mathrm{Attn}(K, V, Q). \qquad (2)$$

The upper part of Fig. 3 visualizes the conventional attention architecture. Note that this attention structure can be used multiple times in one TTS system, by fixing K, V, and setting Q to old attention outputs.

In training, the models converting the text to mel spectrograms are trained to minimize the weighted sum of the mel spectrogram loss [20], and the guided attention loss [20]. The model converting a mel spectrogram to a linear spectrogram is trained to minimized the linear spectrogram loss [20], and loss regarding stop tokens [6].

In synthesis, the model outputs the mel spectrogram in an autoregressive way and outputs the linear spectrogram until a stop token appears.

3. Proposed TTS with multiple language units based on attention

We propose to handle unaligned multiple language units using unit-wise attention architectures. We first propose the architectures and then propose a progressive training strategy to efficiently train our end-to-end TTS.

3.1. TTS using multiple language units

Fig. 3 shows our architecture, which corresponds to "Attentions" in Fig. 2. Linear projected attention scores are calculated in each multiple language unit and summed with weights as fol-

lows.

$$Q + \sum_{u \in U} w_u \mathrm{Attn}(K_u, V_u, Q), \tag{3}$$

where u denotes a symbol of language units, and U denotes the set of u. w_u denotes the weight of each attention score and satisfies

$$\sum_{u \in U} w_u = 1. \tag{4}$$

The attention architectures use the shared Q values, and individual K_u and V_u values to language units. The use of the individual Q values would make parallel calculation difficult, so we incorporate shared Q values.

This structure enables us to train the models with multiple input in an alignment-free way. In other words, although concatenation of multiple inputs at the front-end of TTS requires alignments between the multiple inputs, the unification of the attention outputs in multiple inputs does not because the length of the attention output is the same as that of the Q.

In this paper, in addition to pronunciation ("p") [6, 20], we propose to use subword ("s") as the language units, i.e., $U = p, c, s$. A number of criteria and strategies were proposed [16, 17, 24, 25], and we use language model-based tokenization [17] that is completely language independent.

3.2. Training strategy

We propose a progressive training strategy to train our model with the expectation of archiving stable training. The training starts with a language unit that is dominant for easy-to-predict speech components. Mapping smaller units (e.g., pronunciation) to segmental features of speech can be easily trained. On the other hand, longer units help predict suprasegmental features that are difficult to be trained in the mapping. Thus, our training strategy schedules the weights of attention scores w_u. In the early steps of training, the attention score for only the smallest language unit is used for training, and those of other language units increase during training.

Our training strategy satisfies the following.

1. The weights change exponentially.
2. Only pronunciation is exploited initially.

We exponentially decay the weights of language units other than pronunciation so we can change the weights gradually. We make use of only pronunciation in the former steps so we can sufficiently train the relationship between pronunciation and speech. Without this, we found that segmental features were poorly predicted.

We will explain the design of the progressive training strategy in detail. Let $u_s \in U$ be the smallest unit (pronunciation in this paper). We calculate the weight w_u of the units other than u_s as follows.

$$w_u = \max\left(\frac{1}{|U|}\{1 - \exp\left(-\frac{s - s_{\mathrm{st}}}{d_u}\right)\}, 0.0\right). \tag{5}$$

w_{u_s} is calculated from this formula and Eq. (4). Note that s_{st} is the constant step number when training with the units other than u_s starts, and d_u is the constant value related to the sharpness of the weight curve for unit u. We show the curves of their weights expressed by equations above in Fig. 4. The figure shows two cases; two units(pronunciation and characters), and three units (pronunciation, characters and subwords).

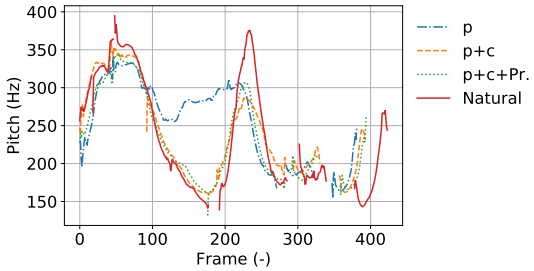

Figure 5: *The example of $F0$ trajectories of the speech synthesized with the evaluation data. "p," "c," "Pr.," and "Natural" indicates pronunciation, character, subword, progressive training strategy, and natural speech, respectively.*

Table 1: *RMSE of log F0. "p," "c," "s," and "Pr." indicate using pronunciation, character, subword, and progressive training strategy, respectively*

p	c	s	Pr.	RMSE
✓				0.223
✓	✓			0.216
✓	✓	✓		0.206
✓	✓		✓	0.216
✓	✓	✓	✓	0.209

4. Experimental evaluation

4.1. Experimental condition

We evaluated our methods with a Japanese end-to-end TTS. We used JSUT [26], a Japanese single-speaker speech corpus, and randomly chose $6,835$ utterances for training and 100 utterances for evaluation. The sampling rate of the natural and synthesized speech was $22,050$ Hz.

We used three language units; pronunciation, character, and subword. We used MeCab [27] to obtain Katakana, Japanese pronunciation symbols, for pronunciation. We used SentencePiece[28] to obtain subword sequences from text. We fixed the size of the subword vocabulary to $4,000$. We trained the subword vocabulary from the text of the training data.

We utilized the DeepVoice3-based implementation [21] as the conventional single input end-to-end model. Our model parameters basically follows those of the implementation. However, we modified the model to some extent to improve the speech quality. The main modification is that we train the mel and linear spectrogram separately as well as DC-TTS [20], even though the original code was designed for training both spectrograms simultaneously.

We fixed the total number of training steps at $450,000$, independent of the methods. We set the starting steps s_{st} to $10,000$. We set the sharpness of the weight curve for subword d_s to $20,000$, and that for character d_c to $40,000$.

4.2. Objective evaluation

As described in **Section 1**, the use of longer units is expected to accurately reproduce the suprasegmental features. To confirm this, we compared log $F0$ of natural and synthesized speech with the root mean squared error (RMSE). Before calculating the RMSE, we used the dynamic time warping algorithm to align the natural and synthesized speech. We used WORLD [29, 30] to extract $F0$. The frame shift length was to set 5 ms.

Table 2: *Results of AB preference tests (w/o progressive training).* ***Bold*** *indicates the preferred model with* p*-value* < 0.05*. "p," "c," "s," and "Pr." indicate using pronunciation, character, subword, and progressive training strategy, respectively*

p	c	s	Pr.	Scores	p-value	p	c	s	Pr.
✓				0.433 vs. **0.567**	$< 10^{-3}$	✓	✓		
✓				0.416 vs. **0.584**	$< 10^{-3}$	✓	✓	✓	
✓	✓			**0.551** vs. 0.449	0.002	✓	✓	✓	

Table 3: *Results of AB preference tests (w/ progressive training).* ***Bold*** *indicates the preferred model with* p*-value* < 0.05*. "p," "c," "s," and "Pr." indicate using pronunciation, character, subword, and progressive training strategy, respectively*

p	c	s	Pr.	Scores	p-value	p	c	s	Pr.
✓	✓			**0.540** vs. 0.460	0.016	✓	✓		✓
✓	✓	✓		0.404 vs. **0.596**	$< 10^{-3}$	✓	✓	✓	✓
✓	✓		✓	0.422 vs. **0.578**	$< 10^{-3}$	✓	✓	✓	✓
✓	✓			0.440 vs. **0.560**	$< 10^{-3}$	✓	✓	✓	✓

Table 1 shows the result. The use of unaligned multiple language units improved the log-$F0$ RMSE compared with a single language unit (i.e., a method using only "p" in the table). Fig. 5 is the example of $F0$ trajectory. We can see the trajectories of "p+c" and "p+c+Pr." drop more clearly around 200th frame than that of "p", although we cannot observe the obvious difference of "p+c" and "p+c+Pr." Thus, although the results do not show any significant improvements with the progressive training, we are confident that the longer units sufficiently contributes to the improvements.

4.3. Subjective evaluation

We conducted preference AB tests on the quality of synthesized speech to check the proposed methods' superiority. We gathered 45 participants for each test via a crowd-sourcing website [31], and each participant evaluated 10 pairs of speech. Note that we created these 10 pairs of speech from 5 different pairs by reversing the order of the proposition.

4.3.1. Single language unit vs. multiple language units

We compared the methods with different language units to see the effect of unaligned multiple language units. The proposed progressive training is not used here, i.e., attentions of language units have the same weights from the start to end of the training.

Table 2 shows the results. We found that the method with one language unit is inferior to that with two or three, and the method with three language unit is inferior to that with two. These are consistent with the results of objective evaluation.

4.3.2. Effect of progressive training strategy

To see the effect of the progressive training, we compared the methods with and without the training. The latter is the same as the method described in **Section 4.3.1**. We also compared other pairs of the multiple language units and the progressive training.

Table 3 shows the results. The method with two language units obtained a higher score without the progressive training strategy. On the other hand, the method with three language units obtained a higher score with the progressive training strategy. The progressive training significantly improves the perceptual naturalness in the case of unaligned three language units. On the other hand, we can see that it is ineffective in the case of two language units. Moreover, as shown in Table 3, we can see that three language units with the progressive training outperforms two language units with and without the progressive training. Therefore we can say that our progressive training can improve naturalness when using three unaligned language units.

Table 4: *Thurstone's paired comparison. "p," "c," "s," and "Pr." indicate using pronunciation, character, subword, and progressive training strategy, respectively*

p	c	s	Pr.	Scores
✓				-0.160
✓	✓			0.061
✓	✓	✓		-0.025
✓	✓		✓	-0.052
✓	✓	✓	✓	0.175

4.3.3. Comparison of all methods

We conducted preference AB tests with all method combinations. Table 2 and Table 3 only show the parts of the results, and here we calculated values of Thurstone's paired comparison using all results to fully evaluate the speech quality.

Table 4 shows the results. All of our methods obtained better scores than with a single language unit, and the use of three language units and the progressive training achieves the best score among all methods. Therefore, we can again say that our methods contribute to improve synthetic speech quality.

5. Conclusion

We proposed an attention-based end-to-end TTS with unaligned multiple language units. This enabled us to use pronunciation symbols, characters, and subwords without aligning them. Also, we designed training schedules to efficiently train the TTS model using multiple language units. Experimental results show that the usage of multiple language units is effective to improve to speech quality. In addition, our proposed progressive training strategy has a good effect on training with subword units. A part of our future work is applying our proposed methods to languages other than Japanese.

6. Acknowledgements

Part of this research and development work was supported by JSPS KAKENHI 18K18100 and 17H06101.

7. 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, pp. 679–682.

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

[3] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "WaveNet: A generative model for raw audio," *arXiv*, vol. abs/1609.03499, 2016.

[4] 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," *arXiv*, vol. abs/1609.03499, 2017.

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

[6] W. Ping, K. Peng, A. Gibiansky, S. Arik, A. Kannan, S. Narang, J. Raiman, and J. Miller, "Deep Voice 3: Scaling text-to-speech with convolutional sequence learning," in *Proc. ICLR*, Vancouver, Canada, Apr. 2018.

[7] J. Sotelo, S. Mehri, K. Kumar, K. K. J. F. Santos, A. C. ville, and Y. Bengio, "Char2Wav: End-to-end speech synthesis," in *Proc. ICLR*, Toulon, France, Apr. 2017.

[8] K. Tokuda, Y. Nankaku, T. Toda, H. Zen, J. Yamagishi, and K. Oura, "Speech synthesis based on hidden Markov models," *Proceedings of the IEEE*, vol. 101, no. 5, pp. 1234–1252, 2013.

[9] H. Z. A. Senior and H. M. Schuster, "Statistical parametric speech synthesis using deep neural networks," in *Proc. ICASSP*, Vancouver, Canada, May. 2013, pp. 7962–7966.

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

[11] J. Fong, J. Taylor, K. Richmond, and S. King, "A comparison of letters and phones as input to sequence-to-sequence models for speech synthesis," in *Proc. SSW*, Vienna, Autria, Sep. 2019, pp. 223–227.

[12] K. Kastner, J. F. Santos, Y. Bengio, and A. Courville, "Representation mixing for TTS synthesis," in *Proc. ICASSP*, Brighton, United Kingdom, May 2019, pp. 5906–5910.

[13] Y. Lu, M. Dong, and Y. Chen, "Implementing prosodic phrasing in Chinese end-to-end speech synthesis," in *Proc. ICASSP*, Brighton, United Kingdom, May 2019, pp. 7050–7054.

[14] J. Zhu, "Probing the phonetic and phonological knowledge of tones in mandarin TTS models," *arXiv*, vol. abs/1912.10915, 2019.

[15] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," *arXiv*, vol. abs/1810.04805, 2018.

[16] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," in *Proc. ACL*, Berlin, Germany, Aug. 2016, pp. 1715–1725.

[17] T. Kudo, "Subword regularization: Improving neural network translation models with multiple subword candidates," in *Proc. ACL*, Melbourne, Australia, Jul. 2018, pp. 66–75.

[18] N. Prateek, M. Łajszczak, R. Barra-Chicote, T. Drugman, J. Lorenzo-Trueba, T. Merritt, S. Ronanki, and T. Wood, "In other news: A bi-style text-to-speech model for synthesizing newscaster voice with limited data," in *Proc. NAACL*, Minnesota, USA, Jun. 2019, pp. 205–213.

[19] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of GANs for improved quality, stability, and variation," *arXiv*, vol. abs/1710.10196, 2017.

[20] H. Tachibana, K. Uenoyama, and S. Aihara, "Efficiently trainable text-to-speech system based on deep convolutional networks with guided attention," in *Proc. ICASSP*, Calgary, Canada, Apr. 2018, pp. 4784–4788.

[21] "Deepvoice3_pytorch," https://github.com/r9y9/deepvoice3_pytorch, accessed: 2020-3-20.

[22] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. NIPS*, California, USA, Dec. 2017, pp. 5998–6008.

[23] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE conference on computer vision and pattern recognition*, Nevada, USA, Jun. 2016, pp. 770–778.

[24] T. Akiyama, S. Takamichi, and H. Saruwatari, "Prosody-aware subword embedding considering Japanese intonation systems and its application to DNN-based multi-dialect speech synthesis," in *Proc. APSIPA*, Hawaii, U.S.A., Nov. 2018, pp. 660–664.

[25] M. Aso, S. Takamichi, N. Takamune, and H. Saruwatari, "Subword tokenization based on DNN-based acoustic model for end-to-end prosody generation," in *Proc. SSW*, Vienna, Austria, Sep. 2019, pp. 234–238.

[26] R. Sonobe, S. Takamichi, and H. Saruwatari, "JSUT corpus: free large-scale Japanese speech corpus for end-to-end speech synthesis," *arXiv*, vol. abs/1711.00354, 2017.

[27] T. Kudo, K. Yamamoto, and Y. Matsumoto, "Applying conditional random fields to Japanese morphological analysis," in *Proc. EMNLP*, Barcelona, Spain, Jul. 2004, pp. 230–237.

[28] T. Kudo and J. Richardson, "SentencePiece," in *Proc. EMNLP*, Brussels, Belgium, Nov. 2018, pp. 66–71.

[29] 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.

[30] M. Morise, "D4C, a band-aperiodicity estimator for high-quality speech synthesis," *Speech Communication*, vol. 84, pp. 57–65, 2016.

[31] "Lancers," https://www.lancers.jp/.