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

Phonetic and Prosodic Information Estimation from Texts for Genuine Japanese End-to-End Text-to-Speech

Naoto Kakegawa[1], *Sunao Hara*[1],*Masanobu Abe*[1],*Yusuke Ijima*[2]

[1]The Graduate School of Interdisciplinary Science and Engineering in Health Systems, Okayama University, Japan
[2]NTT Coporation, Japan

pdef4n86@s.okayama-u.ac.jp, hara@okayama-u.ac.jp, abe-m@okayama-u.ac.jp, ijima@m.ieice.org

Abstract

The biggest obstacle to develop end-to-end Japanese text-to-speech (TTS) systems is to estimate phonetic and prosodic information (PPI) from Japanese texts. The following are the reasons: (1) the Kanji characters of the Japanese writing system have multiple corresponding pronunciations, (2) there is no separation mark between words, and (3) an accent nucleus must be assigned at appropriate positions. In this paper, we propose to solve the problems by neural machine translation (NMT) on the basis of encoder–decoder models, and compare NMT models of recurrent neural networks and the Transformer architecture. The proposed model handles texts on token (character) basis, although conventional systems handle them on word basis. To ensure the potential of the proposed approach, NMT models are trained using pairs of sentences and their PPIs that are generated by a conventional Japanese TTS system from 5 million sentences. Evaluation experiments were performed using PPIs that are manually annotated for 5,142 sentences. The experimental results showed that the Transformer architecture has the best performance, with 98.0% accuracy for phonetic information estimation and 95.0% accuracy for PPI estimation. Judging from the results, NMT models are promising toward end-to-end Japanese TTS.

Index Terms: Text-to-speech, Grapheme-to-Phoneme (G2P), Attention mechanism, transformer, sequence-to-sequence neural networks

1. Introduction

Recently end-to-end text-to-speech (TTS) systems have been intensively studied [1, 2, 3, 4] and showed high potential for generating speech with high quality and naturalness. However, for the construction of genuine end-to-end TTS systems, there are still several problems to be solved. One of them is the front-end text processing including grapheme-to-phoneme (G2P) conversion that generates a sequence of pronunciation symbols (phonemes) given a sequence of letters (graphemes).

In Japanese TTS systems, G2P conversion is more complicated in comparison with English because the Japanese writing system has several types of characters, such as Kanji and Kana characters. Specifically, Kanji characters have multiple corresponding pronunciations that are determined by contexts. Moreover, there is no separation mark between words represented by the sequences of Kanji and Kana characters. In addition to G2P conversion, it is important for Japanese TTS systems to assign an accent nucleus at appropriate positions because Japanese is a pitch-accent language. In short, Japanese TTS systems must have functions for polyphone disambiguation, word segmentation, and accent nuclear shift. In this paper, we call the series of the processing as phonetic and prosodic information (PPI) estimation.

Because of the above reasons, for the end-to-end Japanese TTS systems, PPI is estimated by conventional rule-based or statistical-based modules [5] or is given as a manually annotated source [6, 7]. In this paper, we tackled PPI estimation by DNN (Deep Neural Network) framework, because we would like to construct genuine (fully trainable) Japanese end-to-end TTS systems. We propose to apply NMT models to PPI estimation, more specifically the recurrent neural networks (long short-term memory: LSTM) and the Transformer architecture (Transformer) [8]. Although both are encoder–decoder models with attention mechanism, Transformer could handle the long-range dependencies with self-attention scheme, while LSTM could make best use of the symbol positions in sequence-to-sequence manner. We would like to make sure which models are better for the PPI estimation. In the proposed models, characters like Kanji, Kana, numbers etc. are used as units for input tokens. Because of the character basis approach, the proposed models are free from maintenance of word dictionary for new words, which is expensive in the conventional word-basis systems. In terms of the units for output tokens, we employ syllable-based symbols, because according to previous studies [9, 10], the syllable-based approach is suitable for Japanese TTS prosody controls.

The rest of the paper is organized as follows: Section 2 explains related works and our contributions, Section 3 describes the PPI in detail and the outline of the proposed approach, Section 4 explains the proposed algorithm using the NMT model, Section 5 presents the experiment methods and evaluation results, and Section 6 discusses our conclusions and suggests avenues for future works.

2. Related Works

Because G2P conversion can be viewed as a kind of machine translation, [11] proposed to apply neural machine translation (NMT) approach based on recurrent neural networks to G2P conversion. Thereafter, [12] introduced attention mechanisms [13] combined with recurrent neural networks. Moreover, [14] applied another NMT model of Transformer to G2P conversion and showed better performance than recurrent neural networks with attention mechanisms. In terms of Japanese TTS, accent dictionaries play important role, [15] proposed a neural network-based technique to construct a large vocabulary Japanese accent dictionary, where Kanji and phonetic information are used to estimate the accents of Japanese words. As

Copyright © 2021 ISCA

http://dx.doi.org/10.21437/Interspeech.2021-914

(a) Sentence		富士山は美しい山です															
(b) Kanji-character, Kana-character		富	士	山	は			美	し	い				山	で	す	
Phonetic information	(c) Phonemes in a syllable	/fu/	/ji/	/sa/	/N/	/wa/		/u/	/ts/	/ku/	/shi/	/i/		/ya/	/ma/	/de/	/s/
	(d) Devocalization flag	0	0	0	0	0		0	1	0	0	0		0	0	0	1
Prosodic information	(e) Accent nuclear flag	1	0	0	0	0		0	0	0	1	0		0	1	0	0
	(f)Accent phrase boundary						/						/				
(g) Output symbols of NMT		p1	p2	p3	p4	p5	b1	p7	p8	p9	p10	p11	b1	p12	p13	p14	p15
(h) Graphical representations		ふ’	じ	さ	ん	わ	/	う	つ^	く	し’	い	/	や	ま’	で	す^

Figure 1: *Example of PPI and symbols for NMT*

the front-end text processing, Mandarin TTS has similar problems to Japanese TTS. To solve the problems, [16] proposed to estimate phoneme, tone and prosody from text using sequence-to-sequence approach.

Contributions of this paper would be as follows: (1) The proposed algorithm need only Japanese text to estimate PPIs. Although our approach is similar to [16], Japanese has more polyphone disambiguation and more complicated accent nuclear shift than Mandarin. (2) The proposed algorithm handles texts on the character-basis, not word basis. This enables to estimate PPIs for unknown words that comprise known Kanji characters. (3) As training data, we propose to use a large amount of pairs of sentences and their PPIs that are generated using a conventional Japanese TTS system from 5 million sentences. An aim of the training is to import knowledges from conventional rule-based systems. This is the first step to construct genuine Japanese end-to-end TTS systems based on DNN framework. (4) The performance is evaluated using manually annotated test set, and experiment results show the NMT models can effectively employ context information for Japanese PPI estimation and can successfully estimate PPIs for unknown words.

3. PPI

3.1. PPI estimation in conventional Japanese TTS systems

Figure 2 shows an outline of PPI estimation in conventional TTS systems and Fig. 1 exhibits the examples of PPI for a sentence “富士山は美しい山です（Mt. Fuji is a beautiful mountain).” Fig. 1 (a) depicts that a sentence has no separation mark between words represented by the sequences of Kanji and Kana characters. In conventional TTS systems, morphological analysis is initially performed to segment a sentence into words using a word dictionary. Thereafter, G2P conversion generates phonemes using the word dictionary on a syllable basis as shown in Fig. 1 (c) and devocalization flags (Fig. 1 (d)). It can be observed in the example, Kanji character “山” is converted to different phoneme sequences: /sa//N/ or /ya//ma/. In many cases, a Kanji character may have multiple corresponding pronunciations, which is called a polyphonic character. This makes G2P conversion a hard task in Japanese TTS systems. The series of the above processing generates phonetic information.

After accent phrase boundary is estimated as shown in Fig. 1 (f), an accent nucleus is assigned for each accent phrase as shown in Fig. 1 (e) by referring to accent dictionary and accent nuclear shift rules [17, 18]. Although all the content words have their own accent nucleus position in an accent dictionary, the accent nucleus of an accentual phrase often shifts according to the context, known as accent sandhi. Because Japanese is a pitch-accent language, dealing with the accent sandhi is important to synthesize natural speech, and accent nuclear shift rules are expected to be improved.

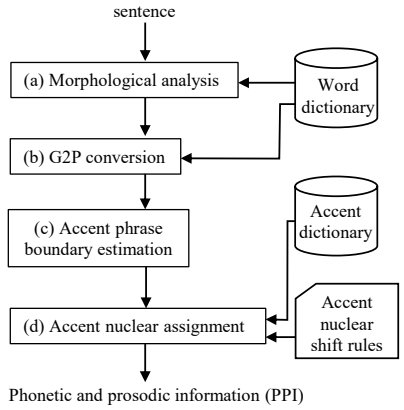

Figure 2: *Outline of PPI estimation in conventional Japanese TTS systems*

3.2. Character-based encoder–decoder approach

In the conventional Japanese TTS systems, the entries of a word dictionary is a Kanji character string（e.g., 日本 (Japan)), a Kana character string（e.g., りんご (apple)), or combinations of them (e.g., 日の出（sunrise)). In the example of Fig. 1, a Kanji character string “富士山（Mt. Fuji)” is searched from the word dictionary. And this dictionary approach is expensive because it needs regular updating to adapt new words. Conversely, the proposed approach is completely different, i.e., the NMT model handles inputs as character-based symbols, such as “富,” “士,” and “山” and generates symbols as shown in Fig. 1 (g). Here, “p1,” “p2,” and “p3” are syllable-based symbols that encode the combination of phonemes in a syllable, devocalization flag, and accent nuclear flag, and “b1” is a symbol that encodes an accent phrase boundary. In the rest of this paper, to explain the proposed method, we call the symbols for input and output as tokens.

In the conventional approach [19], maintaining dictionaries for new words is necessary because PPI estimation is performed

Table 1: *Details of test set annotated manually. Types of numbers and alphabets are large because there are single-byte character set and multi-byte character set.*

Texts	5,142 sentences from newspapers and blog		
		Types	Total number in the dataset
Input tokens	Kana(Hiragana)	80	96,477
	Kana(Katakana)	78	27,783
	Kanji	2,266	110,788
	Numbers	20	11,924
	English alphabets	51	3,124
	Symbols	90	21,366
Output tokens	Syllables	130	288,357
	Accented	133	47,637
	Devocalized	26	14,339
	Without pause	1	38,721
	With short pause	1	14,878
	With long pause	1	17,498

on a word basis. However, the proposed approach deals with a character that is the minimum unit of input text string. Once the NMT model is trained using considerable training data, basic rules might be obtained by the model, and they might work for new words as well.

4. Neural Machine Translation-based Method

The NMT model is a neural network that directly predicts the conditional probability $p(y \mid x)$ of translating a source sentence $(X : x_1, \cdots, x_n)$ to a target sentence $(Y : y_1, \cdots, y_m)$. Here, x_i or y_i represents a word. Moreover, the NMT model consists of two components: (1) an encoder that computes representation X for each source sentence and (2) a decoder that generates one target word of Y at a time. This process is formulated as the following:

$$\log p(y \mid x) = \sum_{j=1}^{m} \log p(y_j \mid y_{<j}, X) \quad (1)$$

In the PPI estimation model, the NMT model translates a Japanese text to a PPI sequence. x_i or y_i is not a word but a token explained Section 3.2. There are three special tokens: <blank> token is a padding token to make the lengths of input and output sequences equal, and <s> and </s> tokens indicate the beginning and the end of sequences, respectively. We proposed the two NMT models for the PPI estimation: (1) LSTM model that consists of Bi-LSTM encoder and LSTM decoder and (2) Transformer model.

5. Experiments

5.1. Dataset preparation for training and evaluation

The training set is generated by a conventional Japanese TTS system [20, 21] from over 5 million newspaper sentences. Although the Japanese TTS system generates PPIs 91% correctly, the generated pairs are regarded as ground truth in the training phase.

We prepared test sets in two ways using the same sentences of 5,142 newspaper and blog sentences: (1) automatic generation and (2) manual annotation. The automatic one is generated by a conventional Japanese TTS system. It is the same way as training set. Using these dataset, we can know how good the MNT models are trained with contaminated training data. Table 1 shows statistics of the kinds of input and output tokens for the manually annotated test set.

Table 2: *Training conditions for LSTM model*

Maximum numbers of tokens in a sentence	300
Encoder	Bi-LSTM
Decoder	LSTM
Encoder layers	2
Decoder layers	2

Table 3: *Training conditions for Transformer*

Maximum numbers of tokens in a sentence	300
Encoder layers	5
Decoder layers	5
Number of attention head	8
Dropout	0.1
Optimizer	Adam
	$\beta_1 = 0.9$
	$\beta_2 = 0.998$
Starting learning rate	2.0
Epoch	40
Loss function	Cross entropy

5.2. Experimental conditions

Table 2 and Table 3 are training conditions for LSTM model and Transformer, respectively. The models are constructed using OpenNMT-py [22], and we mostly use standard options to train the models. We decrease the learning rate if the number of steps becomes large. An input vector is represented by 1-of-K expression. Hence, the dimension of the embedding layer is the same as the vocabulary size.

5.3. Evaluation measure

In terms of the evaluation measure, we employ the concept of word accuracy that is commonly used to evaluate speech recognition systems. Word accuracy is defined as follows:

$$WordAccuracy(\%) = 100 \times (N - S - D - I)/N \quad (2)$$

Here, N is number of words in ground truth. S, D, and I are the numbers of substitutions, deletions, and insertions, respectively. We use tokens instead of words, and S, D, and I are obtained by comparing the estimated sequence of tokens and the ground truth of tokens using dynamic programming (DP).

5.4. Experimental results

5.4.1. Evaluation on the two datasets

We evaluated the proposed models using the test sets of manually annotated and automatically generated. In addition, to know the performance of the conventional TTS system, the automatically generated set is evaluated using the manually annotated set.

Four kinds of accuracy are calculated using Eq. 2: i) P-accuracy that concerns if only phonetic information is correct or not, ii) PP-accuracy that concerns if phonetic and prosodic information are correct or not, iii) B-accuracy is the accuracy of estimation in accent phrase boundaries and calculated for sentences whose all phonetic information is correctly estimated and iv) N-accuracy is the accuracy of estimation in accent nuclear flags and calculated for sentences whose all phonetic information and accent phrase boundaries are correctly estimated.

Table 4 shows the experimental results. As shown in Table

Table 4: *Correct estimation ratio. P-accuracy: accuracy of phonetic information. PP-accuracy: accuracy of PPI. B-accuracy: accuracy of accent phrase boundaries. N-accuracy: accuracy of accent nuclear flags.*

Manually annotated dataset			
	Transformer (proposed)	LSTM	conventional TTS system
P-accuracy	98.0%	98.0%	98.2%
PP-accuracy	95.0%	94.8%	95.3%
B-accuracy	92.8%	91.9%	93.3%
Evaluated sentences	3,046	2,955	3,149
N-accuracy	89.8%	89.6%	89.8%
Evaluated sentences	1,647	1,431	1,774
Automatically generated dataset			
	Transformer (proposed)	LSTM	
P-accuracy	98.2%	98.1%	
PP-accuracy	97.1%	95.9%	
B-accuracy	97.1%	94.6%	
Evaluated sentences	1,007	939	
N-accuracy	98.2%	96.8%	
Evaluated sentences	872	718	

Table 5: *Estimation results for Kanji characters with multiple pronunciations*

		Reference	Estimation
OK	学校へ行った (**went** to school)	イ /i/	イ /i/
OK	運動会を行った (**held** an athletic meet)	オコナ /okona/	オコナ /okona/
OK	長い行列だった (a long **parade**)	ギョー /gyo:/	ギョー /gyo:/
NG	十分かかる (takes **10 minutes**)	ジュップン /juQpuN/	ジューブン /ju:buN/
OK	もう十分だ (**enough**)	ジューブン /ju:buN/	ジューブン /ju:buN/

4, Transformer shows high accuracy from 89.8% to 98.0% for the manually annotated set. Moreover, the performance difference between Transformer and the conventional Japanese TTS system is quite small; less than 0.5%. This indicates that Transformer is successfully trained to estimate PPIs.

In terms of the two proposed models, Transformer outperformed LSTM in all kinds of accuracies for both the automatically generated set and the manually annotated set. Moreover, for both proposed models, the performance for the automatically generated set is slightly better than that for the manually annotated set. This is quite reasonable, because the NMT models are trained using the automatically generated dataset. One of the reasons comes from differences in annotation rule between these datasets. For example, input sequence "言う" is labeled /iu/ in automatically generated dataset, while /yu:/ in manually annotated dataset. For B-accuracy and N-accuracy, the performance degradations are quite larger than PP-accuracy and P-accuracy. These are because accent phrase boundaries and accent nuclear are consistently generated by conventional TTS system, whereas there are variations in manually assignments.

In terms of PP-accuracy and P-accuracy, both models show slightly better performance in P-accuracy than in PP-accuracy. This is reasonable because the types of tokens to be estimated are more in PP-accuracy than in P-accuracy.

5.4.2. Analysis of estimated PPIs

Table 5 shows the estimated examples of Kanji characters that have multiple corresponding pronunciations. Here, "OK" and "NG" indicate that the proposed model estimated the phonetic information correctly and incorrectly, respectively. In the case of Kanji character "行," all of them are correctly estimated such as /i/, /okona/, and /gyo:/ that mean "go," "hold," and "column," respectively. The results indicate that NMT effectively uses context information extracted from characters around "行." On the other hand, in the case of "十分, " the proposed model failed to estimate appropriate phonetic information. This mainly because shortage of the training data. We might need more data that contains "十分."

Table 6: *Estimation results for unknown words*

	Unknown word	Reference	Estimation
OK	爬行	/hako:/	/hako:/
OK	傲岸	/go:gaN/	/go:gaN/
OK	活眼	/katsugaN/	/katsugaN/
NG	嫌厭	/keNeN/	/**iya**eN/

Table 7: *Comparison of the conventional TTS and Transformer*

Word	行った	
Pronunciations	/okonaQta/ (done, held)	/iQta/ (went)
Frequency in the test set	35	6
Accuracy in Transformer	62.8% (22/35)	100% (6/6)
Accuracy in conventional TTS	80.0% (28/35)	100% (6/6)

For the confirmation of the advantages of our character basis modeling instead of word basis used in conventional system, the proposed model is evaluated by estimating phonetic information for unknown words that comprise known Kanji characters. Table 6 exhibits the experimental results, where "OK" and "NG" indicate the same meaning in Table 5. Although Kanji characters have multiple possible phonetic information, the proposed models can estimate correct answers for unknown words. The results indicate that character basis modeling works well. In terms of Table 5 and Table 6, Transformer and LSTM showed the same results.

On the other hand, in some words, the conventional TTS showed slightly better performance for multiple pronunciations. Table 7 shows an example of the comparison of the conventional TTS and Transformer. In the conventional TTS, rules are carefully developed for the words that are frequently used and have strong context dependencies for pronunciations. This suggests more data should be generated by taking contexts into account for the NMT training.

6. Conclusion

In this paper, we proposed to apply NMT for estimating PPI from Japanese texts. Because of the Japanese writing system, PPI estimation is more difficult than G2P conversion for English. The models were trained using pairs of sentences and their PPIs that were generated by the conventional Japanese TTS system from 5 million sentences. In terms of evaluation on automatically generated dataset, the experimental results showed 98.4% accuracy for phonetic information estimation and 97.8% accuracy for PPI estimation. In terms of evaluation on manually annotated dataset, the experimental results showed 98.0% accuracy for phonetic information estimation and 95.0% accuracy for PPI estimation. Moreover, we confirmed that the NMT model can correctly estimate the PPIs of unknown words that comprise known Kanji characters. Judging from the results, the proposed model is promising toward end-to-end Japanese TTS. This is one of the biggest advantages to train the NMT models on the basis of characters.

7. References

[1] Y. Wang, R. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio, Q. Le, Y. Agiomyrgiannakis, R. Clark, and R. A. Saurous, "Tacotron: Towards End-to-End Speech Synthesis," in *Proceedings of 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. Skerrv-Ryan *et al.*, "Natural TTS Synthesis by Conditioning WaveNet on Mel Spectrogram Predictions," in *Proceedings of ICASSP*, 2018, pp. 4779–4783.

[3] W. Ping, K. Peng, A. Gibiansky, S. Ö. Arik, A. Kannan, S. Narang, J. Raiman, and J. Miller, "Deep Voice 3: Scaling Text-to-Speech with Convolutional Sequence Learning," in *Proceedings of ICLR*, 2018.

[4] J. Sotelo, S. Mehri, K. Kumar, J. F. Santos, K. Kastner, A. Courville, and Y. Bengio, "Char2wav: End-to-end speech synthesis," in *Proceedings of ICLR Workshop*, 2017.

[5] M. Suzuki, R. Kuroiwa, K. Innami, S. Kobayashi, S. Shimizu, N. Minematsu, and K. Hirose, "Accent Sandhi Estimation of Tokyo Dialect of Japanese Using Conditional Random Fields," *IEICE Transactions on Information and Systems*, 2017.

[6] 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 *Proceedings of ICASSP*. IEEE, 2019, pp. 6905–6909.

[7] 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 *Proceedings of SSW*, 2019, pp. 166–171. [Online]. Available: http://dx.doi.org/10.21437/SSW.2019-30

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

[9] M. Abe and H. Sato, "Two-stage F0 control model using syllable based F0 units," in *Proceedings of ICASSP*, vol. 2, 1992, pp. 53–56.

[10] H. Nakajima, M. Nagata, H. Asano, and M. Abe, "Estimating Japanese Person Name Accent from Mora Sequence Using Support Vector Machines," *IEICE Trans. Inf. & Syst.(Japanese Edition), D2*, vol. 88, no. 3, pp. 480–488, 2005.

[11] K. Yao and G. Zweig, "Sequence-to-Sequence Neural Net Models for Grapheme-to-Phoneme Conversion," in *Proceedings of Interspeech*, 2015.

[12] S. Toshniwal and K. Livescu, "Jointly learning to align and convert graphemes to phonemes with neural attention models," in *Proceedings of SLT*. IEEE, 2016, pp. 76–82.

[13] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proceedings of ICLR*, 2015.

[14] S. Yolchuyeva, G. Németh, and B. Gyires-Tóth, "Transformer based Grapheme-to-Phoneme Conversion," in *Proceedings of Interspeech*, 2019.

[15] H. Tachibana and Y. Katayama, "Accent Estimation of Japanese Words from Their Surfaces and Romanizations for Building Large Vocabulary Accent Dictionaries," in *Proceedings of ICASSP*, 2020.

[16] J. Pan, X. Yin, Z. Zhang, S. Liu, Y. Zhang, Z. Ma, and Y. Wang, "A Unified Sequence-to-Sequence Front-End Model for Mandarin Text-to-Speech Synthesis," in *Proceedings of ICASSP*, 2020.

[17] Y. Sagisaka and H. Sato, "Accentuation rules for Japanese text-to-speech conversion," *Review of the electrical communication laboratories*, vol. 32, no. 2, pp. 188–199, 1984.

[18] Y. Sagisaka, "Accentuation rules for Japanese word concatenation," *IEICE Trans. Inf. & Syst.(Japanese Edition), D*, vol. 66, no. 7, pp. 849–856, 1983.

[19] T. Kudo, K. Yamamoto, and Y. Matsumoto, "Applying Conditional Random Fields to Japanese Morphological Analysis," in *Proceedings of the Conference on EMNLP*, 2004.

[20] T. Fuchi and S. Takagi, "Japanese Morphological Analyzer Using Word Co-occurrence: JTAG," in *Proceedings of ICCL*, 1998, pp. 409–413.

[21] K. Matsuoka, E. Takeishi, H. Asano, R. Ichii, and Y. Ooyama, "Natural Language Processing In A Japanese Text-to-speech System For Written-style Texts," in *Proceedings of IVTTA*, Sep. 1996, pp. 33–36.

[22] G. Klein, Y. Kim, Y. Deng, J. Senellart, and A. Rush, "OpenNMT: Open-Source Toolkit for Neural Machine Translation," in *Proceedings of ACL*, Jul. 2017, pp. 67–72.