Neural iTTS: Toward Synthesizing Speech in Real-time with End-to-end Neural Text-to-Speech Framework

Tomoya Yanagita[1], *Sakriani Sakti*[1,2], *Satoshi Nakamura*[1,2]

[1]Nara Institute of Science and Technology, Japan
[2] RIKEN, Center for Advanced Intelligence Project AIP, Japan
{yanagita.tomoya.yo8,ssakti,s-nakamura}@is.naist.jp

Abstract

Real-time machine speech interpreters aim to mimic human interpreters that able to produce high-quality speech translations on the fly. It requires all system components, including speech recognition, machine translation, and text-to-speech (TTS), to perform incrementally before the speaker has spoken an entire sentence. For TTS, this poses problems as a standard framework commonly requires language-dependent contextual linguistics of a full sentence to produce a natural-sounding speech waveform. Existing studies of incremental TTS (iTTS) have mainly been conducted on a model based on hidden Markov model (HMM). Recently, end-to-end TTS based on a neural net has synthesized more natural speech than HMM-based systems. In this paper, we take an initial step to construct iTTS based on end-to-end neural framework (Neural iTTS) and investigate the effects of various incremental units on the quality of end-to-end neural speech synthesis in both English and Japanese.

Index Terms: Real-time machine speech interpreters, incremental speech synthesis, end-to-end framework, deep learning

1. Introduction

Speech-to-speech translation (S2ST) is an innovative technology that translates speech signals from a source language to another language, enabling people to communicate with each other by speaking in their own native languages. S2ST systems commonly consist of three components: automatic speech recognition (ASR), machine translation (MT), and text-to-speech (TTS) synthesis. In other words, they first recognize the speech in the source language, automatically translate its words into the other persons language, and finally synthesize them in the target language into speech. In a standard manner, the process is done sentence by sentence. The MT starts to translate to the target text after receiving a complete source sentence from the ASR [1], and TTS synthesizes after receiving the complete target sentence from the MT [2]. However, spoken speech in a lecture or meeting can be very long with unclear sentence breaks. In this case, S2ST will respond with significant delays and creates difficulty for the listeners who are trying to follow the speaker's talk or conversation.

In contrast to the S2ST system, human interpreters generally break sentences into smaller chunks, and incrementally translate based on partial information with minimum delay [3]. But human simultaneous interpreters are expensive, and the range of possible languages that can be translated is usually very limited. Our dream is to construct real-time machine speech interpreters that can imitate the characteristics of the human simultaneous interpreter process. One critical difference with standard S2ST systems is that each component (ASR, MT, TTS) needs to generate the output on the fly before receiving a complete sentence. Several existing works in the ASR and MT fields produce high-quality speech translations while simultaneously minimizing the latency of the translation process. These studies are widely conducted from parametric models to neural network architecture [4, 5, 6, 7, 8, 9, 10, 11]. Unfortunately, research in incremental TTS (iTTS) remains quite limited.

To produce high-quality speech synthesis, many contextual linguistic factors (e.g., phoneme identity and word stress) must be considered because such information can affect the prosodic characteristics of speech. In a standard HMM-based TTS system, the following three processes are typically executed: (a) analyzing the entire sentence and extracting the linguistic features by natural language processing; (b) establishing a sentence-based HMM sequence on the basis of linguistic specifications, and estimating acoustic features while considering the time-series of sentences [12, 13]; (c) reconstructing speech waveform from the predicted acoustic features. In contrast, an iTTS system only has to estimate the target prosody online based on partial knowledge of the syntactic structure of the sentences. The iTTS has to extract the linguistic features in a situation where some linguistic features (the next part of speech tag, the next word, etc.) are unknown during the synthesis. A limited HMM sequence has to be constructed from limited linguistic features, local optimization has to be performed, and acoustic features must then be estimated. Unfortunately, the speech quality is significantly deteriorated due to the limited linguistic features and local optimization.

Several studies attempted to improve the quality by replacing the unknown linguistic features with the average values from a dataset [14], performing a training strategy that can handle the unknown linguistic features [15], and proposing an approach to predict the part of speech of the next word in an acoustic time-series [2, 16, 17]. Although the speech quality can be improved, these existing works have mainly been conducted only with a HMM-based speech synthesis framework. In a pipeline model like HMM-based architecture, all the subcomponents (i.e., linguistic feature extractor, acoustic model, vocoder) are tuned and trained separately, and errors in the earlier stage can propagate through the later stages. Furthermore, the requirement of having full-context labels of linguistic features makes this step difficult for iTTS. Consequently, the speech quality remains limited.

Recently, end-to-end TTS systems have been proposed [18, 19], based on seq2seq with attention [20]. One main factor underlying the popularity of the end-to-end deep-learning architecture is the possibility of simplifying many complicated hand-engineered models and letting DNNs directly map from the character input to the output spaces of speech acoustics. Full context labels in linguistic features are not required anymore, and the synthesized speech quality has outperformed HMM-based frameworks. In this paper, we take an initial step toward constructing a Neural iTTS. To the best of our knowledge, this is the first study that attempts to synthesize speech in real-time using Neural iTTS. We also investigated the effects of various incremental units on the quality of end-to-end neural speech synthesis in both English and Japanese languages.

2. End-to-end TTS

End-to-end TTS tasks model the conditional probability between $p(\mathbf{x}|\mathbf{y})$, where $\mathbf{y} = [y_1, ..., y_T]$ is the sequence of the text input with length T and $\mathbf{x} = [x_1, ..., x_S]$ is the sequence of the (framed) speech features with length S. In this work, the core architecture of an end-to-end TTS is based on Tacotron [18] with several structural modifications. Fig. 1 illustrates our modified Tacotron for English and Japanese, and we describe the difference with the original architecture in the following sections.

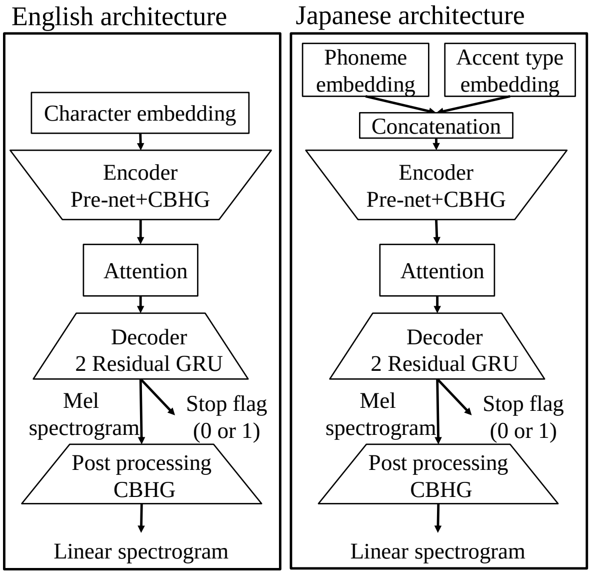


Figure 1: *Architectures of our English and Japanese end-to-end TTS system*

2.1. Overall architecture

The original Tacotron architecture has one encoder and two decoders. The encoder consists of an embedding layer, pre-net layers (two linear layers with dropout) and a CBHG (1-D convolution bank, a highway network, and a bidirectional gated recurrent unit layer) module. At the encoding step, we first project the sequence of text input $\mathbf{y} = [y_1, ..., y_T]$ into the embedding layer, which is fed into two linear layers, and finally we pass the outputs of the full-connected layers into the CBHG components and produce hidden representative $\mathbf{h^e} = [h_1^e, ..., h_T^e]$. On the decoder side, we have two layers of a residual gated recurrent unit (GRU) with attention and post-processing CBHG modules. Given encoder outputs, the attention modules estimate a "context vector" c_s, and then the GRU-decoder estimates log mel-spectrogram $\mathbf{x^M} = [x_1^M, ..., x_S^M]$ from the context vector, and with the pos-processing CBHG, it estimates linear magnitude-spectrogram sequences $\mathbf{x^R} = [x_1^R, ..., x_S^R]$ from all the log mel-spectrogram outputs. Finally, Tacotron reconstructs a sequence of the speech waveform from a linear-spectrogram using the Griffin-Lim algorithm [21].

In this manner, the original Tacotron didn't decide any stopping time-step at the GRU-decoder. To determine the stopping step, we added one linear layer that estimated two values (0: continuation, 1: stop) from the outputs of the residual GRU. The strategy resembles a previously proposed one [19, 22]. The layer uses a sigmoid function as an activation function. For training, we used the following loss function:

$$Loss(\mathbf{x^M}, \mathbf{\hat{x}^M}, \mathbf{x^R}, \mathbf{\hat{x}^R}, \mathbf{s}, \mathbf{\hat{s}}) = \\ (1.0 - \alpha) \times \frac{1}{T} \sum_t^T \{(|x_t^M| - |\hat{x_t}^M|) + (|x_t^R| - |\hat{x_t}^R|)\} \\ -\alpha \times \frac{1}{T} \sum_t^T \{s_t \log(\hat{s_t}) + (1 - s_t) log(1 - \hat{s_t})\}, \quad (1)$$

where $x_t^M, \hat{x_t}^M, x_t^R, \hat{x_t}^R, s_t$, and $\hat{s_t}$ are the truth log mel-spectrogram, the predicted log mel-spectrogram, the truth log magnitude spectrogram, the predicted log magnitude spectrogram, the truth stop flag, and the predicted stop flag at the t frame. The α is a small value (En:1e-7, Ja:1e-5) as a hyperparameter.

2.2. Embedding layer

Specifically to the embedding layer, our English end-to-end TTS follows the original Tacotron that uses character sequences as input to the embedding layers (Fig. 1 left side). But for Japanese, it is difficult because there are three kinds of Japanese characters: hiragana, katakana, and kanji. Since their pronunciation often changes depending on their combinations (especially when combining kanji), the number of model's input may become unwieldy. Here with the current available data, automatically learning the pronunciations is difficult from all the Japanese characters within an end-to-end TTS framework; instead we simply use phoneme sequences as input.

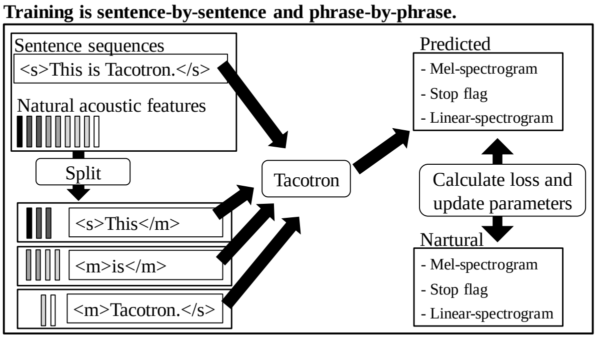
Furthermore, Japanese is a pitch accent language. This means that the meaning of words can be altered by changing the pitch of the same character sequence. For example, "hashi" can mean a bridge or a pair of chopsticks, depending on the pitch accents. However, such pitch information is represented not in words or phonemes but in accent phrase units. Therefore, to accommodate such pitch information, we proposed using the accent type in accented phrases as input. In other words, we use two embedding layer for the phoneme and accent types in a Japanese end-to-end TTS (Fig. 1 right side). After that, we concatenate two embedding outputs as one encoder input.

3. Proposed Neural iTTS

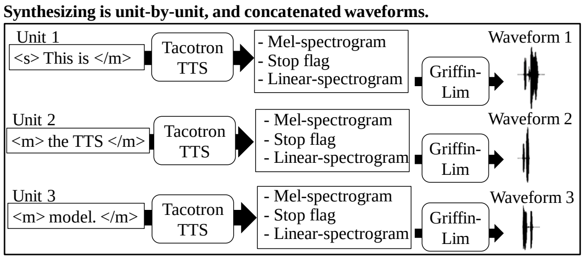
Figure 2 illustrates an overview of the training and synthesizing methods in our proposed incremental end-to-end approach, whose details can be found in the following sub-sections.

3.1. Training and synthesizing strategies in iTTS

Figure 2(a) shows the training process. Since iTTS needs to handle shorter units than sentences, we first prepared our dataset by randomly splitting the full sequences into three parts and then added beginning and end symbols to each input unit text. Here we use different symbols to differentiate the units location within the full sentence: "<s>" as the sentences start, "<m>" as the middle sentences start, "</s>" as the sentences end, and "</m>" as the middle sentences end. However, we still performed the training in a sentence-by-sentence or phrase-by-phrase fashion without much modification to the original one.



(a) Training approach



(b) Synthesizing approach

Figure 2: *Overview of training and synthesizing approaches*

Here we estimated the loss between the natural and estimated features as described in eq. (1), performed back-propagation using this loss, and updated the model parameters.

At the synthesizing stage (Fig. 2(b)), the speech was synthesized using units with a length smaller than a sentence length. Here we stopped the mel-spectrogram outputs by the predicted stop flag ("</s>" end of sentence or "</m>" end of the middle sentence) during the synthesis. After that, we concatenated all the synthesized waveforms of the units into sentence-based speech waveforms. More details about the choice of the synthesized unit can be found in the next subsection.

3.2. Choices of incremental units for iTTS

In existing works in iTTS, most HMM-based iTTS systems were constructed for English, and speech was synthesized word-by-word [2]. In this paper, we investigated several possible synthesized units for an English iTTS:

One word: One incremental unit is one word, so the synthesize process is done word-by-word.

Two words: One incremental unit is two words.

Three words: One incremental unit is three words.

Half sentence: Speech synthesis is done after receiving half of the sentence.

Sentence: Speech synthesis is done after receiving one full sentence. This resembles a non-incremental (standard TTS) which is an upper bound system.

Accent phrase units have been investigated as optimum units in Japanese iTTS systems [17]. Here, we investigated a variant of accent phrase units:

One accent phrase: Since the unit is one accent phrase, the synthesis is done each the accent phrase by the accent phrase.

Two accent phrases: The unit is two accent phrases.

Three accent phrases: The unit is three accent phrases.

Half sentence: Speech synthesis is done after receiving half of the sentence.

Sentence: Speech synthesis is done after receiving one full sentence, and this resembles also a non-incremental TTS.

The smallest incremental unit in English is one word, and the word boundary can be predicted trivially with little latency. Here, we investigate multiple lengths of the smallest unit (i.e., one word, two words, three words, etc.). Half-sentence incremental unit is not predicted. It is an approximation of several words that is more than three words unit. In Japanese, the smallest incremental unit is one accent phrase. It can be predicted with part-of-speech (POS) tag [23]. In real usage, incremental POS tagger to detect an accent phrase is necessary.

In the synthesis and training parts of the TTS system, the first input of the decoder is usually a zero vector of the mel-spectrogram called the "GO frame." We proposed two approaches for connecting the units:

Independent: We assumed that all units are independent and simply used a zero vector in all of them despite the existing previous acoustic features. In this case, the model failed to learn the acoustic time-series within one full sentence.

Look-back context: Except the unit from the beginning of the sentence, for other units that start from the middle of the sentence, we replaced the zero vector with the last vector of the mel-spectrogram from the previous units. In this case, the model may be able to learn the acoustic time-series within one full sentence.

4. Experiment

4.1. English dataset

We used the LJ-speech 1.1 dataset [24]. The original dataset consisted of 13,100 sentences (about 24 hours of speech audio with a 22.05-kHz sampling frequency) spoken by a single female speaker. To get the time-alignment information, we performed forced alignment with the HTS toolkit and obtained about 10-k pairs (speech and text) of successfully generated data. We divided them into 9.8-k pairs of data for training, 100 pairs for the development set, and 100 pairs for the test set. Then as described above, we split each full sentence into three parts with random length inputs and combined these data. The size of training data is four times (full sentence data plus three parts of the divided unit data).

We normalized the text transcription before using it because some words are abbreviations and numerics. The input consists of 43 letters including lower case letters of the alphabet and such special characters as spaces, sentence beginnings/endings, etc. The acoustic features were extracted, and our final set was comprised of 80 dimensions of log mel-spectrogram features, 1024 dimensions of log magnitude spectrogram 80 mel-spectrum, and a 1024-linear spectrum. The frame shift and frame length are 12.5 and 5 ms. The batch size and the optimized method are identical to the original Tacotron. But the reduction factor (the number of predicted frames at one decoding step) is 5 in our model.

4.2. Japanese dataset

We used the JSUT 1.1 dataset that included 7,696 sentences (10 hours of audio sampled at 48-kHz, we downsampled to 22.05-kHz) spoken by a single native female speaker [25]. We used Open Jtalk[1] for extracting the phoneme and accent types from the text. However, Open Jtalk often suffers from incorrect or missing pronunciation. To avoid these mistakes, we used a morphological analysis system (Mecab)[2] and dictionaries for checking mistaken pronunciations. When we got identical pronunciations for both Open Jtalk and Mecab, we added them to the data. We also removed the "Repeat 500" sub-dataset (Dataset was recorded to 100 transcriptions five times). Finally, we had 5276 pairs (speech and text) of data and divided them into 5-k data pairs for training, 100 pairs for the development set, and 100 pairs for the test set.

The input text consists of 45 phoneme symbols and 20 accent types. Size of each embedding layer is half size of English Tacotron. The other parameters are used under the same conditions as for the English.

4.3. Subjective evaluation of prosodic quality with or without the context from previous units

As described earlier, we proposed two approaches for connecting the units: (1) "Independent" and (2) "Look back context." We used the zero vector or the last vector from the previous input in the initial input vector of each unit in the decoder and performed an A/B preference test on these two approaches. For the evaluation, we concatenated the outputs of each incremental synthesized speech into one utterance of a pseudo sentence. Consequently, the resulting utterances may have an unnatural prosodic connection. Evaluators listened to the two synthesized speeches of full sentences without knowing whether those speech utterances were generated through the incremental procedure with the context from previous units or not. After that, they were requested to choose the sample with more natural prosodic-connecting quality of synthesized units within one full sentence. This evaluation was conducted with ten native Japanese speakers for Japanese Neural iTTS, and ten English speakers who have TOEIC ip[3] score higher than 700 points (daily conversational level) for English Neural iTTS. There were 40 speech utterances (20 utterances per method), which were presented in random order. An A/B-test was performed that differentiated among three preference: (1) the first synthesized speech has better natural prosodic quality, (2) the second synthesized speech has better natural prosodic quality, or (3) no difference. Each speech utterance could be played as many times as the subjects wished. Fig. 3 shows the result of the Japanese A/B-test. The results show the system with the second approach ("Looking-back context") that used the log mel-spectrogram inputs from previous units had better natural prosodic quality than only using the zero vectors. This means that the quality improved when the model considered the acoustic time-series. Based on this result, we selected this second approach for further Japanese evaluation. On the other hand, the result of the English AB-test shown in Fig. 4 reveals that the system with the first approach ("Independent") that used the zero vector had better natural prosodic quality than only using the ero vectors. This means that the quality improved when the model may not consider the acoustic time-series, otherwise this occurred poor stop flag prediction. Table 1 shows a mean of evaluation loss corresponding to predict stop flag. As can be seen, the English iTTS has poor prediction than the Japanese model. This means that method with "Look-back context" may cause a lousy estimation, and the error may propagate to the next synthesis unit. Based on this result, we selected this first approach "Independent" for further evaluation of the English system.

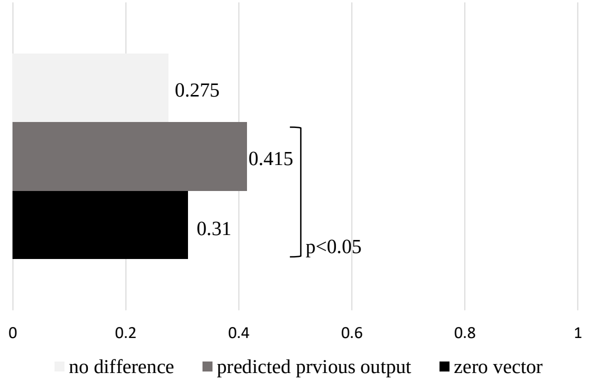


Figure 3: *A/B preference test of Japanese prosodic quality with or without the context from previous units in the initial input.*

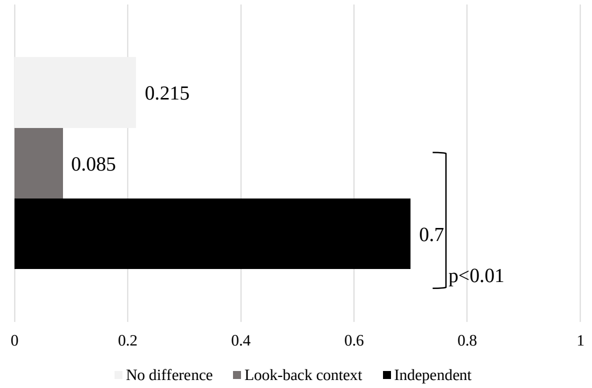


Figure 4: *A/B preference test of English prosodic quality with or without the context from previous units in the initial input.*

Table 1: *A mean value of development loss for stop flag prediction*

	Japanese	English
Independent	0.0115	0.0422
Look-back context	0.0108	0.0385

4.4. Subjective evaluation of naturalness on the effect of different lengths of the incremental unit

Next we conducted a mean opinion score (MOS) test as a subjective evaluation for naturalness on the effect of different lengths of incremental units. Subjects listened to each presented speech audio and rated the overall quality based on its naturalness. A 5-point MOS scale was used, where 5 indicated excellent speech utterances (very clear and completely

[1] Open Jtalk – http://open-jtalk.sourceforge.net/

[2] Mecab – https://taku910.github.io/mecab/

[3] Test of English for International Communication

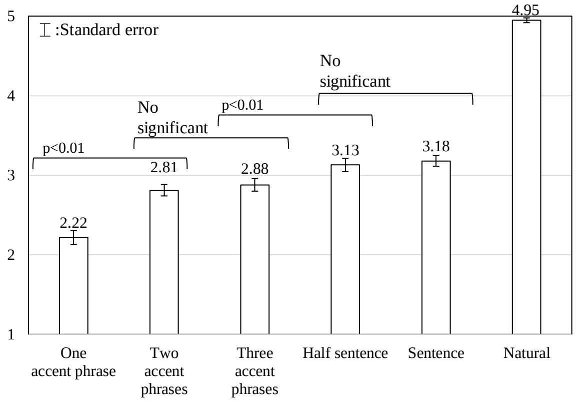


Figure 5: *Subjective evaluation for naturalness of Japanese synthesized speech with various lengths of the synthesized unit.*

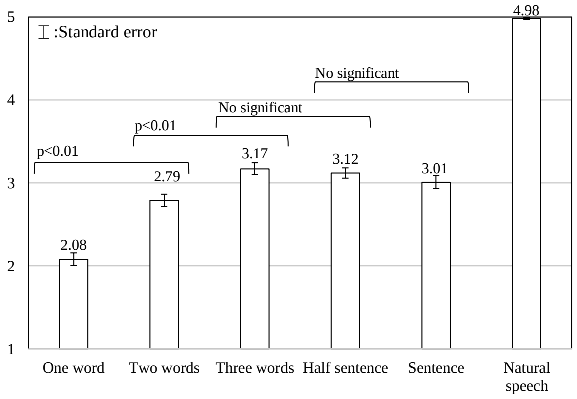


Figure 6: *Subjective evaluation for naturalness of English synthesized speech with various lengths of the synthesized unit.*

natural sounds) and 1 indicated bad speech utterances (unclear and completely unnatural sounds). We conducted the subjective evaluations in Japanese with ten native speakers and in English with ten speakers under the same conditions of English A/B test. Here we have five iTTS systems with various lengths of incremental units: one accent phrase, two accent phrases, three accent phrases, half sentences, and full sentences. With natural speech, there were six types of synthesized speech to be evaluated. 78 speech utterances (13 utterances per synthesized unit) were presented in random order. Each speech utterance could be played as many times as the subjects wished.

Figure 5 and Figure 6 show the naturalness of the MOS scores at each synthesized unit in Japanese and English Neural iTTS respectively. Since we do not use a wavenet vocoder[26], there has been a wide gap between generated speech and natural speech. Furthermore, it is natural that a shorter unit has a larger decrease in quality. But note the size of the effect in MOS quality due to different unit lengths. The shortest unit length (one accent phrase or one word) reached almost two points, and the synthesized speech quality improved from the one unit to connecting two/three units (see the increasing MOS score between two and three units). It is surprising that the scores with half sentence units resemble the full sentence units (non-incremental) in Japanese case. Also in the case of English, the scores with three words resemble the full sentence units (non-incremental). The reason might be because we train only a single model with full sentence data plus three parts of the divided unit data for incremental and non-incremental cases. Consequently, as the number of shorter unit data is larger than the complete sentence data, the model may have a bias to the shorter units. In the future, we will investigate this phenomenon in more details. Towards development of real-time machine speech interpreters, the results suggest to use Japanese Neural iTTS with incremental synthesized units for between the three accent phrases to the half-sentence units, and English Neural iTTS with incremental synthesized units between two and three words.

5. Conclusions

This paper presents the first end-to-end neural iTTS system. Specifically, we proposed end-to-end neural iTTS architecture with training and synthesis strategies to handle partial information in real-time situations. To consider acoustic time-series, we also proposed the initial input of each unit to be the last vector of the Mel-spectrogram from the previous unit. The experiments have been done on Japanese and English datasets. We found that using “Look-back context” approach could improve the prosodic naturalness between synthesized units in Japanese. Moreover, we explored the effect of various lengths of synthesized units in the MOS quality for Japanese and English end-to-end iTTS. Our result reveals that a synthesized unit between three accent phrases and half-sentences suggests an optimal synthesized unit in end-to-end Japanese iTTS, while English synthesized unit is shorter. In the future, we will further investigate the performance of Neural iTTS, given the partial output from the MT systems in a full-pledge real-time machine speech interpreters.

6. Acknowledgements

Part of this work was supported by JSPS KAKENHI Grant Numbers JP17H06101 and JP17K00237.

7. References

[1] E. Matusov, A. Mauser, and H. Ney, “Automatic sentence segmentation and punctuation prediction for spoken language translation,” in *in Proc. IWSLT*, 2006, pp. 158–165.

[2] M. Pouget, O. Nahorna, T. Hueber, and G. Bailly, “Adaptive latency for part-of-speech tagging in incremental text-to-speech synthesis,” in *17th Annual Conference of the International Speech Communication Association (Interspeech 2016)*, 2016, pp. 2846–2850.

[3] F. Goldman-Eisler, “Segmentation of input in simultaneous translation,” *Journal of psycholinguistic Research*, vol. 1, no. 2, pp. 127–140, 1972.

[4] C. Fügen, A. Waibel, and M. Kolss, “Simultaneous translation of lectures and speeches,” *Machine Translation*, vol. 21, no. 4, pp. 209–252, Dec 2007.

[5] S. Bangalore, V. K. Rangarajan Sridhar, P. Kolan, L. Golipour, and A. Jimenez, “Real-time incremental speech-to-speech translation of dialogs,” in *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, ser. NAACL HLT ’12, Stroudsburg, PA, USA, 2012, pp. 437–445.

[6] T. Fujita, G. Neubig, S. Sakti, T. Toda, and S. Nakamura, “Simple, lexicalized choice of translation timing for simultaneous speech translation,” in *14th Annual Conference of the International Speech Communication Association (InterSpeech 2013)*, Lyon,

France, August 2013, pp. 3487–3491. [Online]. Available: http://www.phontron.com/paper/fujita13interspeech.pdf

[7] H. Shimizu, G. Neubig, S. Sakti, T. Toda, and S. Nakamura, "Constructing a speech translation system using simultaneous interpretation data," in *10th International Workshop on Spoken Language Translation (IWSLT)*, Heidelberg, Germany, December 2013, pp. 212–218. [Online]. Available: http://www.phontron.com/paper/shimizu13iwslt.pdf

[8] Y. Oda, G. Neubig, S. Sakti, T. Toda, and S. Nakamura, "Optimizing segmentation strategies for simultaneous speech translation," in *The 52nd Annual Meeting of the Association for Computational Linguistics (ACL)*, Baltimore, USA, June 2014, pp. 551–556. [Online]. Available: http://www.phontron.com/paper/oda14acl.pdf

[9] N. Jaitly, Q. V. Le, O. Vinyals, I. Sutskever, D. Sussillo, and S. Bengio, "An online sequence-to-sequence model using partial conditioning," in *Advances in Neural Information Processing Systems 29*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, Eds. Curran Associates, Inc., 2016, pp. 5067–5075. [Online]. Available: http://papers.nips.cc/paper/6594-an-online-sequence-to-sequence-model-using-partial-conditioning.pdf

[10] J. Gu, G. Neubig, K. Cho, and V. O. Li, "Learning to translate in real-time with neural machine translation," in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*. Valencia, Spain: Association for Computational Linguistics, Apr. 2017, pp. 1053–1062. [Online]. Available: https://www.aclweb.org/anthology/E17-1099

[11] T. N. Sainath, C.-C. Chiu, R. Prabhavalkar, A. Kannan, Y. Wu, P. Nguyen, and Z. Chen, "Improving the performance of online neural transducer models," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5864–5868.

[12] K. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi, and T. Kitamura, "Speech parameter generation algorithms for HMM-based speech synthesis," in *Acoustics, Speech, and Signal Processing, 2000. ICASSP'00. Proceedings. 2000 IEEE International Conference on*, vol. 3. IEEE, 2000, pp. 1315–1318.

[13] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Simultaneous modeling of spectrum, pitch and duration in HMM-based speech synthesis," in *Sixth European Conference on Speech Communication and Technology*, 1999.

[14] T. Baumann, "Decision tree usage for incremental parametric speech synthesis," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 3819–3823.

[15] M. Pouget, T. Hueber, G. Bailly, and T. Baumann, "Hmm training strategy for incremental speech synthesis," in *16th Annual Conference of the International Speech Communication Association (Interspeech 2015)*, 2015, pp. 1201–1205.

[16] T. Baumann and D. Schlangen, "Evaluating prosodic processing for incremental speech synthesis," in *Thirteenth Annual Conference of the International Speech Communication Association*, 2012.

[17] T. Yanagita, S. Sakti, and S. Nakamura, "Incremental TTS for Japanese language," *Proc. Interspeech 2018*, pp. 902–906, 2018.

[18] Y. Wang, R. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio, Q. Le, Y. Agiomyrgiannakis, R. Clark, and R. A. Saurous, "Tacotron: Towards end-to-end speech synthesis," in *Proc. Interspeech 2017*, 2017, pp. 4006–4010. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2017-1452

[19] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan, R. A. Saurous, Y. Agiomvrgiannakis, and Y. Wu, "Natural tts synthesis by conditioning wavenet on mel spectrogram predictions," 04 2018, pp. 4779–4783.

[20] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[21] D. Griffin and J. Lim, "Signal estimation from modified short-time fourier transform," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 32, no. 2, pp. 236–243, 1984.

[22] A. Tjandra, S. Sakti, and S. Nakamura, "Listening while speaking: Speech chain by deep learning," in *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2017, pp. 301–308.

[23] 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*, vol. 100, no. 4, pp. 655–661, 2017.

[24] K. Ito, "The LJ speech dataset," https://keithito.com/LJ-Speech-Dataset/, 2017.

[25] R. Sonobe, S. Takamichi, and H. Saruwatari, "Jsut corpus: free large-scale japanese speech corpus for end-to-end speech synthesis," *arXiv preprint arXiv:1711.00354*, 2017.

[26] A. Van Den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. W. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio." *SSW*, vol. 125, 2016.