10th ISCA Speech Synthesis Workshop
20-22 September 2019, Vienna, Austria

Rakugo speech synthesis using segment-to-segment neural transduction and style tokens — toward speech synthesis for entertaining audiences

Shuhei Kato[1,2], *Yusuke Yasuda*[1,2], *Xin Wang*[2], *Erica Cooper*[2], *Shinji Takaki*[3], *Junichi Yamagishi*[1,2,4]

[1]SOKENDAI (The Graduate University for Advanced Sciences), Japan
[2]National Institute of Informatics, Japan [3]Nagoya Institute of Technology, Japan
[4]The University of Edinburgh, UK

{skato,yasuda,wangxin,ecooper,jyamagis}@nii.ac.jp, takaki@sp.nitech.ac.jp

Abstract

We have been working on constructing *rakugo* speech synthesis as a challenging example of speech synthesis that entertains audiences. *Rakugo* is a traditional Japanese form of verbal entertainment that is similar to one-person stand-up comedy. In rakugo, a performer himself/herself plays multiple characters, and conversations by them make the story progress. We tried to build a rakugo synthesizer with state-of-the-art encoder-decoder models with attention such as Tacotron 2. However, it did not work well because the expressions of rakugo speech are far more diverse than those of read speech. We therefore use segment-to-segment neural transduction (SSNT) in place of a combination of attention and decoder. Furthermore, we experimented with global style tokens (GST) and manually-labeled context features to enrich the speaking styles of synthesized rakugo speech. The results show that SSNT greatly helps align the encoder and decoder time steps and that GST help reproduce characteristics better.

Index Terms: rakugo, speech synthesis, Tacotron, SSNT, style tokens

1. Introduction

The quality of synthetic speech has been drastically improved, and some systems have achieved the same mean opinion scores (MOS) as those of natural speech, albeit under limited conditions, as reported in [1, 2]. Well-articulated read speech is commonly used for speech synthesis (text-to-speech; TTS) research and products. Recently, modeling speech with various speaking styles has also been actively studied in deep-learning-based speech synthesis [3, 4, 5].

Most of us would agree that the main function of speech would be communication and information transfer, in other words, media. Namely, speech transfers its contents, the emotions of speakers, personality of speakers, intention of speakers, etc. to listeners and previous studies on speech synthesis mainly aimed to achieve such basic functions accurately.

However, speech has a function beyond simple communication and information transfer. For example, verbal entertainment, including *rakugo*, which is a traditional Japanese form of verbal entertainment that is similar to one-person stand-up comedy, entertains audiences through the medium of speech. In other words, speech in verbal entertainment has an important role in stirring listeners' emotions.

Why do audiences enjoy such speech in verbal entertainment? One answer is that the content is just funny. But that's not everything. For example, when a new performer and an experienced professional perform an identical rakugo story, the audience would likely enjoy the latter's performance much more. It is obvious that how speech is spoken has an important effect upon the listener.

Toward speech synthesis systems that can entertain audiences, we have been working on building a rakugo speech synthesizer [6] because rakugo is a good and challenging example of verbal entertainment in Japan. Our goal is to construct speech synthesis systems that can entertain audiences and to prove that TTS can have a new function beyond simple communication and information transfer.

We see that some speech-synthesis-based rakugo works, mostly including a lot of manual intervention, have already been submitted to online video platforms [7, 8, 9]. Whether you can enjoy such works would depend on your view or subjectivity, but we think the quality of these works are far poorer than that of works of professional rakugo performers at various levels. As far as we know, there is no machine-learning-oriented fundamental research on modeling rakugo speech for speech synthesis.

We therefore first recorded and built a large rakugo speech database ourselves for our experiments because most commercial rakugo recordings are live recordings that contain noise and reverberation, and no suitable rakugo speech databases usable for speech synthesis exist.

Some readers may wonder how different rakugo speech is compared with that of audiobooks, which is an active research topic in the speech synthesis field. Of course they are closely related to each other, but we think that the main difference is that almost all parts of rakugo consist of conversations and dialogues of characters that are performed from memory. Rakugo speech is therefore *casually-pronounced, monodramatic, and conversational data* whereas audiobooks contain well-pronounced monological read speech.

Using the database, we first tried to use Tacotron 2 [1] to synthesize rakugo speech because it is reported that it can model audiobooks as well as read speech. However, we could not successfully model rakugo speech using Tacotron 2[1]. That would be partially or mostly because expressions of rakugo speech are far more diverse than those of read speech as described earlier. We therefore introduce a more robust encoder-decoder model called segment-to-segment neural transduction (SSNT) in place of a combination of attention and decoder [11, 12]. Furthermore, we combine global style tokens (GST) [3] or manually labeled context embeddings with it to enrich the speaking styles of synthesized rakugo speech.

The rest of this paper is structured as follows: Section 2 introduces an overview of rakugo. Section 3 describes the details of our rakugo speech database. Section 4 introduces our

[1]Our implementation of Tacotron 2 has obtained very high MOS for read speech in both Japanese [10] and English.

10.21437/SSW.2019-20

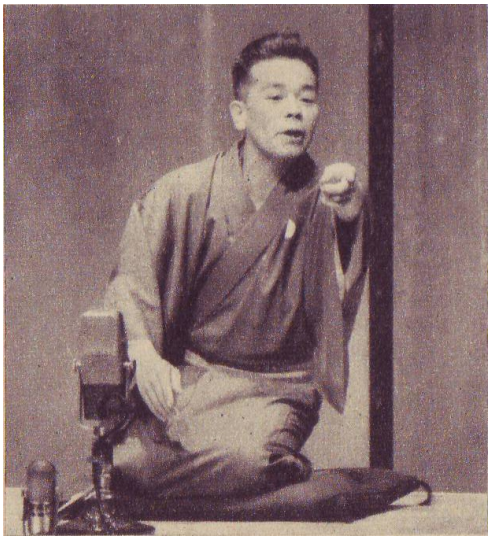

Figure 1: *Sanyutei Sansho, who was a professional rakugo player, is performing rakugo [13].*

SSNT-based speech synthesis. Section 5 gives the experimental conditions, results, and discussion. Section 6 concludes with our findings and future work.

2. Rakugo

Rakugo is a traditional Japanese form of one-person verbal entertainment. It has over three hundred years of history, and it is generally divided into *Edo* (Tokyo) rakugo and *Kamigata* rakugo, which has been developed in Osaka and Kyoto. A rakugo performer, who is called *hanashika* if he/she is a professional, sits down on a cushion and performs improvisationally or from memory alone on a stage (Figure 1). He/she plays multiple characters by himself/herself, and their conversations make the story progress. In the main part of a story, to be mentioned later, almost all of the parts consist of conversations. No properties other than a fan and a hand towel are used.

Rakugo performers tell stories in their performance. A rakugo story consists of a *maeoki* (greeting), *makura* (introduction), the main part, *ochi* (punch line), and *musubi* (conclusion) [14]. Maeoki is not necessary, and musubi appears in stories without ochi[2], or is used when performers discontinue stories because of time limitations. Makura is often improvised. Ochi, the punch line, is the most important part of rakugo (the meaning of the word "rakugo" is "a story with ochi").

Rakugo stories are generally divided into standards, which were established by about the 1920s, and modern stories, which were created after the 1930s. In this paper, we deal with standards. It should be noted that the Japanese used in standards is slightly old-fashioned.

The following is an example of a rakugo paragraph.

> Tomi: Whoa! Oh no! Oh no! Oh no! Oh no!
> Friend: Wait Tomi. What are you doing?
> Tomi: Oh, I'm chasing after a thief.
> Friend: Seriously? Aren't you the fastest man in this town? He is unlucky.
> Friend: Which direction did he escape?
> Tomi: He's catching up with me.

[2]Rakugo stories generally have ochi, but exceptionally, some stories such as *kaidanbanashi* (ghost stories), do not have ochi.

3. Database

We built a rakugo speech database for our experiments because there is no rakugo speech database suitable for speech synthesis. Most commercial rakugo recordings are live recordings that include noise and reverberation; therefore, we recorded the rakugo speech ourselves.

The performer was Yanagiya Sanza, who is a professional rakugo performer with over 20 years' experience. Only he was in the recording booth, and he did not face an audience nor receive any reaction from one. He performed 25 Edo (Tokyo) rakugo standards. We retook no recordings on account of mispronunciation or restatements except in cases where the performer asked us to do so.

The first author transcribed pronunciation of the recorded speech. We defined no special symbols for mispronunciation, fillers, or laughs. We used a comma only at a pause in a sentence. We used a dot at the end of a sentence, and used a question mark at the end of a sentence that ends with rising intonation. The ratio of the number of question sentences, which have question marks at the end of sentences, and that of others is about 3:7. We separated sentences according to the following criteria.

- A place we can separate sentences grammatically followed by a pause.
- A place where a turn-taking occurs.
- A place right after a rising intonation.

We also added context labels to each sentence (Table 1). All the labels excluding **part** are defined by the first author because no well-known categories of them exist in rakugo.

We think the **role** of the character is important because almost all of speech in rakugo, especially in the main part, is conversations of characters. The **individuality** of the character is a special category for fool characters, usually called *Yotaro*, who often appear in rakugo stories. We think the **condition** of characters is also important because characters speak in various speaking styles. All of the styles are defined by the first author via carefully listening to speech and reading contexts. The **relationship** of the companions to talk with is defined because in conversations in rakugo (of two characters), one must be dealt with as the superior and the other as the inferior. **N_companion** (number of companions to talk with) is defined because characters may talk to himself/herself, or speak to one person or multiple persons. The **distance** to the companions to talk with is defined because characters may speak to someone near or far from themselves. In the context of the **part** of the story, we considered *maeoki* (greetings) and *musubi* (conclusion) as *makura* (introduction) and *ochi* (punch line), respectively.

4. SSNT-based speech synthesis

4.1. SSNT-based speech synthesis

Our rakugo speech synthesis uses segment-to-segment neural transduction (SSNT) based text-to-speech synthesis (TTS) [12]. The SSNT-based TTS is an encoder-decoder model that can input a text or phoneme sequence and output a variable length of a mel spectrogram, but unlike all other encoder-decoder TTS [1, 15, 16, 17, 18], the SSNT-based TTS does not use the soft attention mechanism. This is because the attention mechanism is too flexible. Context vectors of the attention network are allowed to use input information at any time step of the encoder network. If the attention network is not well-trained, this results

Table 1: *Context labels ("hanashika" refers to improvised or narrative speech in makura)*

Group	Name	Details
ATTRibute of the character	**role** of the character	**sex**: hanashika, male, female; **age**: hanashika, child, young, middle-aged, old; **rank**: hanashika, *samurai*, artisan, merchant, other townsperson, countryperson, with other dialog, modern, other
	individuality of the character	hanashika, fool
CONDition of the character	**condition** of the character	neutral, admiring, admonishing, affected, angry, begging, buttering up, cheerful, complaining, confident, confused, convinced, crying, depressed, drinking, drunk, eating, encouraging, excited, fearing, feeling sketchy, feeling sick, feeling sleep, feeling sorry, feeling suspicious, find it easier than expected, freezing, frustrated, ghostly, happy, hesitating, interested, justifying, *kakegoe*, loud voice, laughing, leaning on, lecturing, looking down, panicked, pet directed speech, playing dumb, putting up with, rebellious, refusing, sad, seducing, shocked, shouting, small voice, soothing, straining, surprised, swaggering, teasing, telling off, tired, trying to remember, underestimating, unpleasant
SITuation of the character	**relationship** to the companions to talk with	hanashika, narrative, soliloquy, superior, inferior
	n_companion: number of companions to talk with	hanashika, narrative, soliloquy, one, two or more
	distance to the companions to talk with	hanashika, narrative, near, middle, far
STRucture of the story	**part** of the story	makura, main part, ochi

in unacceptable errors such as skipping input words, repeating the same phrases, and prolonging the same sounds.

Since the speech synthesis database normally has speech data and corresponding text aligned well, it is reasonable to have more explicit constraints. The SSNT-based TTS uses the explicit constraints. In SSNT [11], the decoder is allowed to consider the following two alignment options only: 1) stay at the same encoder time step and increment the decoder time step and 2) transit to the next encoder time step and increment the decoder time step. It then computes a joint probability of an output feature sequence and the left-to-right self-transition alignment. The above motivation is very similar to the hidden Markov model, but SSNT uses neural networks to compute decoder and alignment probabilities in a nonlinear way on the basis of the outputs of the encoder network.

The overall network structure of the SSNT-based TTS used in our experiments is shown in Figure 2. A letter or phoneme one-hot vector is converted into an embedding, and is input to three convolutional layers, followed by a bidirectional LSTM, the same process as Tacotron 2.

The output of the encoder is expected to have non-linearly encoded linguistic information. This is passed onto the decoder network and concatenated with the information obtained from auto-regressive feedback. The predicted acoustic feature at the previous time step is fed back to a prenet, which acts as an information bottleneck, and then is processed with a stack of LSTM layers. LSTM further considers contextual information of the feedback, and the output of the LSTM layers is concatenated with the output of the encoder network. This is further transformed via feedforward layers and is used as the basis of alignment and of emission networks. The alignment network predicts the above two transition probabilities via the sigmoid layer, and the emission network predicts the mel spectrum at the current decoder time step using feedforward layers followed by a linear output layer. During the training, we use a forward-backward algorithm and optimize the network. During the synthesis, we use greedy decoding and generate speech. For more details, please refer to [12].

4.2. SSNT-based speech synthesis with global style tokens

Moreover, we use "global style tokens" (GST). The GST framework is a prosody transfer approach for the encoder-decoder TTS [3]. It first extracts prosody from reference audio via a reference encoder and and then creates a style embedding vector, which will be propagated to the decoder network. This can be easily integrated into the above SSNT-based TTS framework to enrich the speaking styles of synthesized rakugo speech. Like the original paper, the reference encoder and the style token layer convert an input reference mel spectrogram into a fixed-length style embedding [3]. The style embedding vector is constant within a sentence. It is concatenated with the phoneme embedding at the encoder network and is then propagated to the decoder network.

5. Experiments

5.1. Experimental conditions

We used 16 of all 25 stories in the database because it is work in progress. They are about 4.3 hours long and contain 7,337 sentences. They did not include very short (< 0.5 s) or very long (≥ 20 s) speech. We used 6,459 sentences for training, 717 sentences for validation, and 161 sentences for testing. It should be noted that the total amount of speech was rather small. We tried to build Tacotron-based models with a larger amount of speech containing the very short or very long speech above, and we also tried to fine-tune from well-trained Tacotron-based models trained with read speech [19], but their quality was similar to or

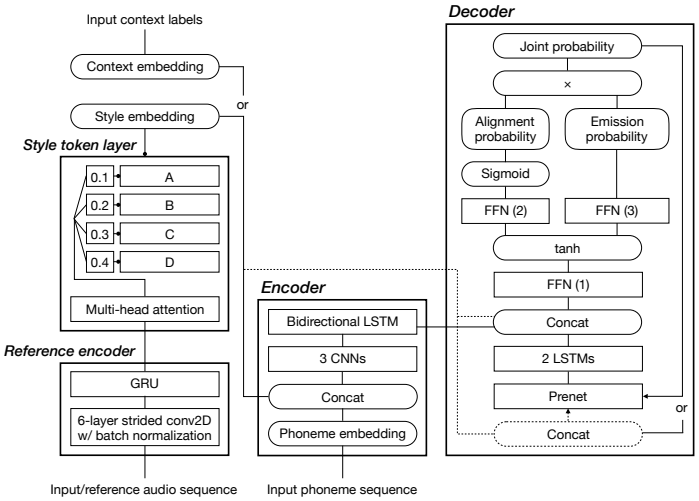

Figure 2: *The overall network structure of the SSNT-based TTS. Dashed lines refer to SSNT-ATTR+ and SSNT-context+.*

worse than those that produced the results below.

We trained several models for our experiments.

- **SSNT** is an SSNT model, and no style embedding or contexts is input.
- **SSNT-GST-*n*** is a GST-based SSNT model, and the number of multi-head attention in the style token layer is n. We used 4, 8, 16, 32, and 64 as n. It should be noted that the reference audio of the test sets is natural speech itself.
- **SSNT-ATTR** is a context-embedding-based SSNT model, and input context is ATTR (role and individuality) only.
- **SSNT-context** is a context-embedding-based SSNT model, and all the contexts are input.
- **SSNT-ATTR+** and **SSNT-context+** are the same models as SSNT-ATTR and SSNT-context, respectively, except for the additional concatenation of context embeddings with encoder outputs and feedback to the decoder (dashed lines shown in Figure 2).
- **Tacotron**, **Tacotron-ATTR**, and **Tacotron-context** are the same models as SSNT, SSNT-ATTR, and SSNT-context, respectively, but Tacotron 2 with postnet is used instead of SSNT. We used forward attention with a transition agent, which would align better than the originally used location sensitive attention as reported in [20].

In the case of GST-based models, a reference mel spectrogram and a phoneme sequence were used as inputs. We used natural (ground-truth) audio as reference audio when predicting test speech. In the case of context-embedding-based models, a phoneme sequence and a context embedding are used as inputs. The context embeddings are constant within a sentence.

We trained each model with about 3,500 epochs. We used zoneout regularization at every LSTM [21]. Input mel spectrograms are converted from waveforms normalized to -26 dBov by sv56 in each sentence [22]. Values of mel spectrum are transformed into 0 mean and 1 standard deviation at each dimension over the whole data. Other configurations are shown in Table 2.

Predicted mel spectrograms are converted into waveforms by a WaveNet vocoder trained with natural mel spectrograms and waveforms of all the training, validation, and test sets [23, 24, 25]. The sampling rate of the waveform was 16 kHz.

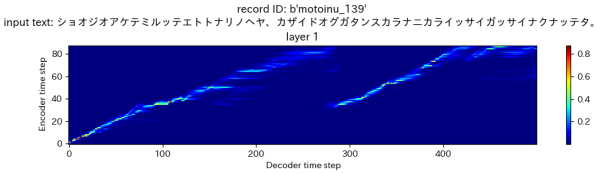

Figure 3: *An example of obvious alignment errors.*

5.2. Results and discussion

5.2.1. Alignment errors

As objective evaluation, we first compared the rates of obvious alignment errors of the Tacotron and SSNT systems. We used test sentences and first generated an alignment probability of each of the input phoneme sequences. We then computed the expected values of the alignment probability over decoder time steps. This tells us which phoneme input within a sentence is used at each encoder time step in general. Because the phoneme sequence should be used from the beginning to the end sequentially in order, we can easily tell that an obvious error happens if adjacent encoder time steps have a large gap (e.g. four time steps or more) or if the next encoder time step is not a monotonic increase (e.g. minus two time steps). Moreover, we count it as an error if the last encoder time step does not correspond to the last decoder time step.

Alignment error rates per system are shown in Table 3. The results show that the SSNT systems greatly reduced the alignment errors. We think this is because alignment transition in SSNT is rather more restricted than that of soft attention mechanism. Additional concatenation of context embeddings with encoder outputs and feedback to the decoder seems to reduce alignment errors further. Figure 3 is an example of obvious alignment errors caused by the Tacotron system.

5.2.2. Listening test

We selected a set of sentences consisting of a short story as materials for our listening test. A total of 161 sentences consisting of 12 stories were prepared for the listening test, and sentence-level short audio files were concatenated as one long audio file per story. Because the speech samples were synthesized sentence by sentence, and pauses between sentences were not predicted, the pauses between sentences we used were based

Table 2: *Configurations and hyperparameters used for our SSNT-based TTS systems*

Sampling rate	48 kHz
FFT length	4096
Spectral analysis	Frame length: 50 ms; frame shift: 12.5 ms; window: Hann
Number of mel filters	80
Phoneme embedding	512 dims
Style embedding	512 dims
Context embedding	4 dims (ATTR), 68 dims (context)
Reference encoder conv2D	3×3 filters with 2×2 stride, SAME padding, ReLU activation; number of filters: $128, 128, 256, 256, 512, 512$
Reference encoder GRU	512-unit unidirectional
Number of style tokens	10
Encoder CNN	512 units, 5×1 filters, ReLU activation, 50% drop rate
Encoder LSTM	512-d cells, 10% zoneout rate
Decoder prenet	2 fully-connected layers of 256 ReLU units, 50% drop rate
Decoder LSTM	1024-d cells, 10% zoneout rate
Decoder FFN	(1) 2 fully-connected layers of 256 tanh units, (2) 1 sigmoid unit, (3) 80 linear units
Reduction factor	2
Optimization	Adam
Learning rate	Initial learning rate: 0.0001, exponential decay
Size of mini-batch	32 (SSNT), 96 (Tacotron)

Table 3: *Alignment error rates*

System	Alignment error rate (%)
SSNT	1.86
SSNT-GST-4	2.48
SSNT-GST-8	1.86
SSNT-GST-16	3.11
SSNT-GST-32	1.86
SSNT-GST-64	3.73
SSNT-ATTR	2.48
SSNT-ATTR+	**1.24**
SSNT-context	1.86
SSNT-context+	**1.24**
Tacotron	28.57
Tacotron-ATTR	29.19
Tacotron-context	26.09

on real audio recordings. This should be predicted by systems, but that would be be out of the scope of this paper. Listeners evaluated speech NOT in sentence-by-sentence samples but in a whole (short) story. Natural speech recordings are not included in our test because we wanted to see the differences between the systems more precisely instead of the differences between TTS and natural speech.

We conducted a MOS test. In each evaluation, listeners listened to the same short story generated using one of the models listed in 5.1 in each screen and evaluated its 1) naturalness, 2) whether they could distinguish each character, and 3) whether they could understand the content, using the five-point MOS scale for each. A total of 135 listeners conducted 453 evaluation rounds in all. The results are shown in Figure 4.

First we can see that the proposed SSNT systems have higher scores than the Tacotron systems in all three questions. Among the SSNT systems, SSNT-GST-8 slightly outperformed the others. GST can increase the quality of speech itself and its representation, but too many heads of multi-head attention seem to reduce quality. This is probably because of overfitting. Context features did not increase the quality, and too many contexts may also drop the quality as SSNT-context and SSNT-context+ did, also probably because of overfitting. The scores of question 2) and 3) show similar trends. A better distinction of each character and a better understanding of the content may be correlated with each other.

6. Conclusion

We have built a rakugo speech synthesizer using a new SSNT-based TTS framework. SSNT greatly reduced alignment errors compared to Tacotron 2, and we were able to train a reasonable model even from highly expressive speech. From the listening test results, we observed that the use of GST helped improve the quality of speech, but the use of context features did not. We plan to make further improvements by using a combination of GST and context features and by modeling pauses between sentences.

Acknowledgements This work was partially supported by a JST CREST Grant (JPMJCR18A6, VoicePersonae project), Japan, and by MEXT KAKENHI Grants (16H06302, 17H04687, 18H04120, 18H04112, 18KT0051), Japan. The numerical calculations were carried out on the TSUBAME 3.0 supercomputer at the Tokyo Institute of Technology.

7. References

[1] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerry-Ryan, R. A. Saurous, Y. Agiomyrgiannakis, and Y. Wu, "Natural TTS synthesis by conditioning WaveNet on mel spectrogram predictions," in *Proc. ICASSP*, 2018, pp. 4779–4783.

[2] N. Li, S. Liu, Y. Liu, S. Zhao, and M. Liu, "Neural speech synthesis with transformer network," in *Proc. AAAI*, 2019.

[3] Y. Wang, D. Stanton, Y. Zhang, R. Skerry-Ryan, E. Battenberg, J. Shor, Y. Xiao, F. Ren, Y. Jia, and R. A. Saurous, "Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis," arXiv:1803.09017 [cs.CL], 2018.

[4] W.-N. Hsu, Y. Zhang, R. J. Weiss, H. Zen, Y. Wu, Y. Wang, Y. Cao, Y. Jia, Z. Chen, J. Shen, P. Nguyen, and R. Pang, "Hierarchical generative modeling for controllable speech synthesis," arXiv:1810.07217 [cs.CL], 2018.

[5] T. Wood, "Varying speaking styles with neural text-to-speech," Alexa Blogs, 2018. [Online]. Available: https://developer.amazon.com/zh/blogs/alexa/post/7ab9665a-0536-4be2-aaad-18281ec59af8/varying-speaking-styles-with-neural-text-to-speech

[6] S. Kato, S. Takaki, J. Yamagishi, Y. Yasuda, and X. Wang, "Use and evaluation of Tacotron and context features in rakugo speech synthesis (in Japanese)," in *Technical Report of IEICE*, vol. 118, no. 497, 2019, pp. 161–166.

[7] MSS, "Hanashika Miku no tokusen rakugo Manju kowai desu (in Japanese)," 2009. [Online]. Available: https://www.nicovideo.jp/watch/sm5899050

[8] Metsuki-warui-P, "[VOCALOID rakugo] Kamban no pin [Metsuki warui Miku] (in Japanese)," 2011. [Online]. Available: https://www.nicovideo.jp/watch/sm13959846

[9] zky, "[Hatsune Miku] VOCALOID rakugo "Nozarashi" (in Japanese)," 2012. [Online]. Available: http://www.nicovideo.jp/watch/sm17066984

[10] Y. Yasuda, X. Wang, S. Takaki, and J. Yamagishi, "Investigation of enhanced Tacotron text-to-speech synthesis systems with self-attention for pitch accent language," in *Proc. ICASSP*, 2019, pp. 6905–6909.

[11] L. Yu, J. Buys, and P. Blunsom, "Online segment to segment neural transduction," in *Proc. EMNLP*, 2016, pp. 1307–1316.

[12] Y. Yasuda, X. Wang, and J. Yamagishi, "Initial investigation of an encoder-decoder end-to-end TTS framework using marginalization of monotonic hard latent alignments," submitted to SSW10, 2019.

[13] The Asahi Shimbun Company, Ed., *The Asahi Picture News, June 28 1959 issue (in Japanese)*. The Asahi Shimbun Company, 1959, p. 20.

[14] M. Nomura, *Rakugo No Gengogaku (in Japanese)*. Kodansha, 2013.

[15] J. Sotelo, S. Mehri, K. Kumar, J. F. Santos, K. Kastner, and A. C. Courville, "Char2Wav: End-to-end speech synthesis," in *Proc. ICLR*, 2017.

[16] Y. Wang, R. J. 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, pp. 4006–4010.

[17] Y. Taigman, L. Wolf, A. Polyak, and E. Nachmani, "VoiceLoop: Voice fitting and synthesis via a phonological loop," in *Proc. ICLR*, 2018.

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

[19] H. Kawai, T. Toda, J. Yamagishi, T. Hirai, J. Ni, N. Nishizawa, M. Tsuzaki, and K. Tokuda, "XIMERA: A concatenative speech synthesis system with large scale corpora," *IEICE Transactions on Information and System (Japanese Edition)*, pp. 2688–2698, 2006.

[20] J.-X. Zhang, Z.-H. Ling, and L.-R. Dai, "Forward attention in sequence-to-sequence acoustic modeling for speech synthesis," in *Proc. ICASSP*, 2018, pp. 4789–4793.

[21] D. Krueger, T. Maharaj, J. Kramár, M. Pezeshki, N. Ballas, N. R. Ke, A. Goyal, Y. Bengio, A. Courville, and C. Pal, "Zoneout: Regularizing RNNs by randomly preserving hidden activations," in *Proc. ICLR*, 2017.

[22] International Telecommunication Union, Recommendation G.191: Software tools and audio coding standardization, 2005.

[23] X. Wang, J. Lorenzo-Trueba, S. Takaki, L. Juvela, and J. Yamagishi, "A comparison of recent waveform generation and acoustic modeling methods for neural-network-based speech synthesis," in *Proc. ICASSP*, 2018, pp. 4804–4808.

[24] A. Tamamori, T. Hayashi, K. Kobayashi, K. Takeda, and T. Toda, "Speaker-dependent WaveNet vocoder," in *Proc. INTERSPEECH*, 2017, pp. 1118–1122.

[25] A. van den 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:1609.03499 [cs.SD], 2016.

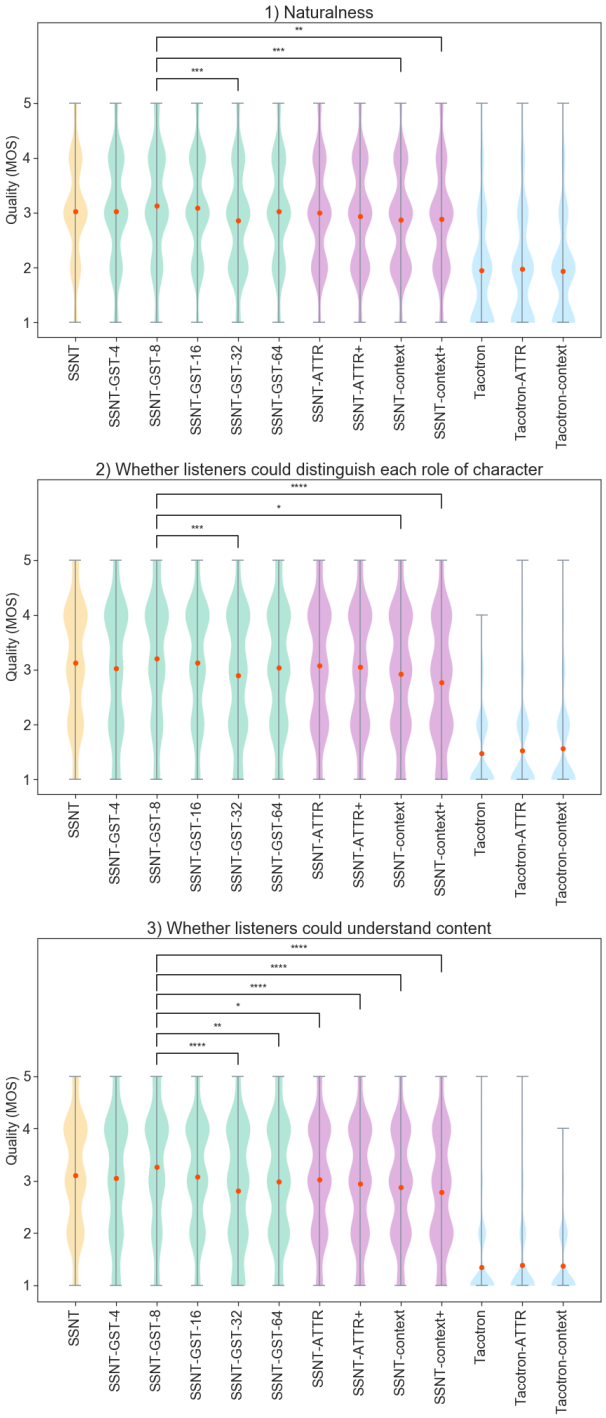

Figure 4: *Results of the listening test.* *: $p < 0.05$, **: $p < 0.01$, ***: $p < 0.005$, ****: $p < 0.001$. *Only significant differences via Brunner-Munzel test with Bonferroni correction between SSNT-GST-8 and the other SSNT systems are shown. There are significant differences between each SSNT system and each Tacotron system ($p < 0.001$).*