INVESTIGATION OF ENHANCED TACOTRON TEXT-TO-SPEECH SYNTHESIS SYSTEMS WITH SELF-ATTENTION FOR PITCH ACCENT SPEECH LANGUAGE

Yusuke Yasuda1,3, Xin Wang1, Shinji Takaki1, Junichi Yamagishi1,2,3*

1National Institute of Informatics, Japan 2The University of Edinburgh, Edinburgh, UK 3SOKENDAI (The Graduate University for Advanced Studies), Japan

yasuda@nii.ac.jp, wangxin@nii.ac.jp, takaki@nii.ac.jp, jyamagis@nii.ac.jp

ABSTRACT

End-to-end speech synthesis is a promising approach that directly converts raw text to speech. Although it was shown that Tacotron2 outperforms classical pipeline systems with regards to naturalness in English, its applicability to other languages is still unknown. Japanese could be one of the most difficult languages for which to achieve end-to-end speech synthesis, largely due to its character diversity and pitch accents. Therefore, state-of-the-art systems are still based on a traditional pipeline framework that requires a separate text analyzer and duration model. Towards end-to-end Japanese speech synthesis, we extend Tacotron to systems with self-attention to capture long-term dependencies related to pitch accents and compare their audio quality with classical pipeline systems under various conditions to show their pros and cons. In a large-scale listening test, we investigated the impacts of the presence of accentual-type labels, the use of force or predicted alignments, and acoustic features used as local condition parameters of the Wavenet vocoder. Our results reveal that although the proposed systems still do not match the quality of a top-line pipeline system for Japanese, we show important stepping stones towards end-to-end Japanese speech synthesis.

Index Terms— speech synthesis, deep learning, Tacotron

1. INTRODUCTION

Tacotron [1] opened a novel path to end-to-end speech synthesis. It enables us to directly convert input text to audio. Unlike traditional pipeline methods that typically consist of separate text analyzer, acoustic, and duration models, Tacotron handles everything as a single model, which reduces laborious feature engineering and error propagation across cascaded models. Indeed, Tacotron2, which is a combination of the Tacotron system and WaveNet [2], successfully generated audio signals that resulted in very high MOS scores comparable to human speech [3].

The above achievements of Tacotron and Tacotron2 and similar results reported for Clarinet [4], and Transformer based TTS [5] are confirmed only for English, and there have been only a few investigations into such architectures with other languages to the best of our knowledge. This is partially or mainly because additional challenges must be overcome for other languages. This study focuses on the Japanese language, which is among the most challenging languages.

Japanese writing has three types of orthographical characters: Hiragana, Katakana, and Kanji (Chinese). The diversity of characters in Japanese causes a critical problem related to rare characters. Moreover, Japanese is a pitch-accented language, and accentual-types (accent nucleus positions) may change the meanings of words. However, accentual-types are not explicitly shown in Japanese characters. Moreover, due to the accent sandhi phenomena, accent nucleus positions are context dependent, so they change positions depending on adjacent words. Because of these problems, state-of-the-art systems for Japanese are dominantly pipeline systems that still rely on an external text analyzer including hand-written dictionaries and rules of pitch accent types for each word or word-to-accentual-type predictors trained on such external resources [6]. An end-to-end approach may potentially simplify these processes in a data-driven way.

Towards the development of end-to-end Japanese TTS systems, we apply the Tacotron system to the Japanese language. We first propose enhanced systems with self-attention to capture long-term dependency better. We then compare their audio quality with that of classical pipeline systems under various conditions. Finally, we conduct a large-scale listening test to investigate the impacts of the presence of accentual-type labels, the use of force- or predicted alignments, and acoustic features used as local condition parameters of the Wavenet vocoder.

The remaining part of this paper is structured as follows. In Section 2, we describe our Japanese Tacotron systems enhanced with self-attention. Section 3 shows experimental conditions and the results of a large-scale listening test. Section 4 concludes with our findings and our future work.

2. PROPOSED ARCHITECTURES FOR JAPANESE TTS

2.1. Tacotron using phoneme and accentual type

In this section, we describe our slightly modified baseline Tacotron [1] that can handle Japanese accentual-type labels. We refer to this system as JA-Tacotron. Figure 1-A shows its architecture. Tacotron is a sequence-to-sequence architecture [2] that consists of encoder and decoder networks. Unlike classical pipeline systems with explicit duration models, Tacotron uses an attention mechanism [8] that implicitly learns alignments between source and target sequences. In this paper, we use phoneme and accentual-type sequences as a source and mel-spectrogram as a target as our first investigation towards end-to-end Japanese speech synthesis. This baseline architecture is inspired from [9], which applied Tacotron to the Chinese language. On the encoder side, phoneme and accentual-type sequences are embedded to separate embedding tables with different dimensions, and the embedding

---

*This work was partially supported by JST CREST Grant Number JPMJCR18A6, Japan and by MEXT KAKENHI Grant Numbers (16H06302, 17H04687, 18H04120, 18H04112, 18KT0051), Japan.
The self-attention block consists of self-attention, followed by a fully connected layer with tanh activation and residual connection. We use multi-head dot product attention [12] as an implementation of self-attention. This block is inserted after LSTM layers at the encoder and decoder. At the encoder, the output of CBH-LSTM layers is processed with the self-attention block. Since LSTM can capture the sequential relationships of inputs, we do not use positional encoding [5]. Both self-attended representation and the original output of the CBH-LSTM layers are final outputs of the encoder.

At the decoder, the two outputs from the encoder are attended with a dual source attention mechanism [13]. We choose a different attention mechanism for each source, forward attention for the output of CBH-LSTM and additive attention for the self-attended values. This is because we want to utilize the benefits of both: forward attention accelerates alignment construction, and additive attention provides flexibility to select long-term information from any segment. In addition, we can visualize both alignments. Unlike the encoder, self-attention with 5 ms latency is directly applied to the decoder. At each time step of decoding, the self-attention layer attends all past frames of LSTM outputs and outputs only the latest frames as a prediction output. The predicted frames are fed back as input for the next time step.

2.3. Tacotron using vocoder parameters

Explicitly modeling the fundamental frequency ($F_0$) might be a more appropriate choice for TTS systems for pitch-accent languages. To incorporate $F_0$ into the proposed systems, we further developed a variant of SA-Tacotron by using vocoder parameters as targets. We use mel-generalized cepstrum coefficients (MGC) and discretized log $F_0$ as vocoder parameters, and we predict these parameters with Tacotron. We choose 5 ms for the frame shift to extract MGC and $F_0$ as such fine-grained analysis conditions are typically required for reliable speech analysis based on vocoders. However, note that this condition is not a natural choice for training Tacotron, which typically uses coarse-grained condition, usually 12.5 ms frame shifts and 50 ms frame lengths, to reduce input and output mismatch. With a frame shift of 5 ms, the length of target vocoder parameter sequences becomes 2.5 times longer than the normal 12.5 ms condition. In other words, 2.5 times longer autoregressive loop iteration is required to predict a target, so this task is much more challenging. To alleviate the difficulty, we set the reduction factor to be three in order to reduce the target length. This setting results in 5/3 times longer target length compared to SA-Tacotron in the previous section.

Figure 1-C shows the modified architecture of the SA-Tacotron using MGC and log $F_0$ as targets. To handle the two types of vocoder parameters, we introduce two pre-nets and three output layers at the decoder. The output layers include a MGC prediction layer that uses MGC and $F_0$ as such fine-grained analysis conditions are typically required for reliable speech analysis based on vocoders. We

At training time, since all target frames are available, this computation can be parallelized by applying a step mask. Since the decoder depends on LSTM, the whole computation cannot be parallelized, but this optimization decreases memory consumption because all past LSTM outputs do not need to be preserved at each time step to calculate gradients on a backward path in backpropagation algorithm. Thanks to this optimization, we can train the extended architecture with a negligible increase in training time.

We tried larger reduction factors, but the audio quality deteriorated as the reduction factor increased.

2

3

At training time, since all target frames are available, this computation can be parallelized by applying a step mask. Since the decoder depends on LSTM, the whole computation cannot be parallelized, but this optimization decreases memory consumption because all past LSTM outputs do not need to be preserved at each time step to calculate gradients on a backward path in backpropagation algorithm. Thanks to this optimization, we can train the extended architecture with a negligible increase in training time.

We tried larger reduction factors, but the audio quality deteriorated as the reduction factor increased.
represented discretized \( \log F_0 \) as one-hot labels at training time, but feed back predicted probability values at inference time \[14\]. We use L1 loss for MGC and cross entropy error for discretized \( \log F_0 \) and stop flag, and we optimize the model by using the weighted sum of the three losses. The cross entropy error of \( \log F_0 \) is scaled by 0.45 to adjust its order to the other two loss terms.

3. EXPERIMENTS

3.1. Experimental conditions

We used a Japanese speech corpus from the ATR Ximera dataset \[15\]. This corpus contains 28,959 utterances from a female speaker and is around 46.9 hours in duration. The linguistic features, such as phoneme and accentual-type label, were manually annotated, and the phoneme label had 58 classes, including silence, pause, and short pause \[16\]. To train our proposed systems, we trimmed the beginning and ending silence from the utterances, after which the duration of the corpus was reduced to 33.5 hours. We used 27,999 utterances for training, 480 for validation, and 142 for testing.

For the experiment, we built several TTS systems as listed in Table 1. The JA-Tacotron and SA-Tacotron with and without accentual-type labels were built to show whether the investigated architectures can learn lexical pitch accents in an unsupervised manner. We also built a SA-Tacotron that uses vocoder parameters instead of mel-spectrogram as the acoustic features. In addition, we included JA-Tacotron with forced alignment instead of predicted alignment to understand the accuracy of duration modeling better. With forced alignment, alignments are calculated with teacher forcing, and target acoustic parameters are predicted with the alignments obtained with teacher forcing. Note that, in this setting, even though forced alignments are calculated with teacher forcing, acoustic parameter prediction itself does not use teacher forcing.

For JA-Tacotron and SA-Tacotron, we allocated 32 dimensions for accentual-type embedding and 224 dimensions for phoneme embedding. For the models without accentual-type embedding, 256 dimensions were allocated to phoneme embedding. We set the reduction factor to be two for the models using mel-spectrogram as a target and three for the models using vocoder parameters. All the predicted frames of the acoustic features were fed back as the next input. At inference time, the inference was stopped on the basis of a binary stop flag as in \[14\]. The network was optimized with Adam optimizer \[17\]. We used exponential learning decay with an initial rate 0.0005 for the models using mel-spectrogram, and 0.002 for the models using vocoder parameters. We implemented our proposed systems using TensorFlow\(^4\).

For baseline systems, we included two classical pipeline systems that use vocoder parameters and mel-spectrogram \[16\], \[18\], \[19\]. Unlike the architecture of our proposed systems, these pipeline systems used full context labels as linguistic features and needed to have duration prediction models. To test how the accuracy of duration prediction affects the naturalness of synthetic speech, we compared phoneme duration predicted by a hidden semi-Markov model (HSMM) with oracle alignments obtained by force alignments. Finally, as a reference for how much listeners are sensitive to accentual phrase boundaries obtained by forward attention.

GMC and \( F_0 \) (vocoder parameters). These two WaveNets had the same network structure as that in our previous study \[19\].

3.2. Objective evaluation

**What does self-attention learn?:** Figure 2 shows a visualization of the attention layers of SA-Tacotron learned on the Japanese corpus. The first figure from the top shows the alignment of an encoder LSTM source and target mel-spectrogram (forward attention). Bottom figure shows alignment between output of encoder’s self-attention block and target mel-spectrogram (additive attention). Vertical white lines indicate accentual phrase boundaries obtained by forward attention.

**Comparison of mel-spectrogram and vocoder parameters:** The alignment between source phoneme and target spectrogram frames should monotonically increase. Non-monotonic alignment may result in mispronunciation, some phonemes being skipped, repetition, the same phoneme continuing, and intermediate termination. We therefore manually counted abnormal alignment errors included in the test set. We observed no alignment errors for JA-Tacotron and SA-Tacotron using mel-spectrograms as a target. However, alignment errors were found for SA-Tacotron using vocoder parameters due to the longer length than the corresponding mel-spectrogram. We found 19 alignment errors out of 142 test utterances.

3.3. Subjective evaluation

We recruited 236 native Japanese speakers as listeners by crowdsourcing. The listeners evaluated 32 samples from 16 systems in a single test set. This includes natural speech and analysis by synthesis (copy synthesis). One listener can evaluated at most 10 test sets. One sample was evaluated 20 times and we got 45,440 data points in total. Figure 4 shows five-point mean opinion scores.

---

\(^4\) The source codes is available at https://github.com/nii-yamagishilab/self-attention-tacotron

\(^5\) This system is named MOC in \[16\].
Fig. 3: Natural mel-spectrogram (top figure), mel-spectrogram predicted from SA-Tacotron with accentual-type labels (middle figure), and mel-spectrogram predicted from SA-Tacotron without labels (bottom figure). Black arrow in the bottom figure points wrong harmonics that results in wrong accent. White lines show accentual phrase boundaries acquired from attention’s output.

Table 1: TTS systems used for our analysis. Notations are \( V \): vocoder parameters, \( M \): mel spectrogram, \( S \): accentual type label, \( N \): no accentual type label, \( P \): predicted alignment, \( F \): forced alignment.

<table>
<thead>
<tr>
<th>System</th>
<th>Architecture</th>
<th>Acoustic feature</th>
<th>Accent label</th>
<th>Alignment</th>
</tr>
</thead>
<tbody>
<tr>
<td>SATMAP</td>
<td>SA-Tacotron</td>
<td>MGC &amp; ( F_0 )</td>
<td>predicted</td>
<td>predicted</td>
</tr>
<tr>
<td>SATMAP</td>
<td>SA-Tacotron</td>
<td>Mel-spec. 12.5 ms</td>
<td>predicted</td>
<td>predicted</td>
</tr>
<tr>
<td>SATMNP</td>
<td>SA-Tacotron</td>
<td>Mel-spec. 12.5 ms</td>
<td>N/A</td>
<td>predicted</td>
</tr>
<tr>
<td>TACMNF</td>
<td>JA-Tacotron</td>
<td>Mel-spec. 12.5 ms</td>
<td>predicted</td>
<td>force-aligned</td>
</tr>
<tr>
<td>TACMNF</td>
<td>JA-Tacotron</td>
<td>N/A</td>
<td>force-aligned</td>
<td>predicted</td>
</tr>
<tr>
<td>TACMPF</td>
<td>Pipeline</td>
<td>MGC &amp; ( F_0 )</td>
<td>predicted</td>
<td>force-aligned</td>
</tr>
<tr>
<td>TACMPF</td>
<td>Pipeline</td>
<td>MGC &amp; ( F_0 )</td>
<td>corrupted</td>
<td>force-aligned</td>
</tr>
<tr>
<td>TACMQF</td>
<td>Pipeline</td>
<td>Mel-spec. 5 ms</td>
<td>predicted</td>
<td>force-aligned</td>
</tr>
<tr>
<td>TACMPF</td>
<td>Pipeline</td>
<td>Mel-spec. 5 ms</td>
<td>N/A</td>
<td>predicted</td>
</tr>
</tbody>
</table>

of the proposed and baseline systems for the listening test results. Statistical significance was analyzed using the two-sided Mann-Whitney statistical test.

**What is the effect of accentual-type labels?:** All proposed systems without accentual-type labels got significantly lower scores than the corresponding systems with labels; for example, JA-Tacotron without labels had a score of 2.63 ± 0.03 whereas JA-Tacotron with labels got 3.46 ± 0.03. This means that the architectures of the proposed systems cannot learn lexical pitch accents in an unsupervised fashion and require additional inputs. The pipeline system with corrupted labels also showed a significant drop with a score of 3.27 ± 0.03. This shows that incorrect accents affected listener’s judgments towards the naturalness of the synthetic speech.

**Does self-attention help?:** SA-Tacotron had better scores than JA-

Tacotron for each condition with or without accentual-type labels. This indicates that self-attention layers have a positive effect on the naturalness. Among our proposed systems, SA-Tacotron with labels (SATMAP) got the highest score of 3.60 ± 0.03.

**Comparison of mel-spectrogram and vocoder parameters:** SA-Tacotron using vocoder parameters got a relatively low score, 2.99 ± 0.03, even if it used accentual-type labels and self-attention layers. This is because this system generated alignment errors due to the prediction of longer sequences as we described in the previous section. Among the baseline systems, the systems using MGC and \( F_0 \) had higher scores than the systems using mel-spectrogram under both the forced and predicted alignment conditions.

**Comparison of predicted and forced alignment:** Interestingly, JA-Tacotron using forced alignment got lower scores than that using predicted alignment under both conditions with and without accentual-type labels. This result is surprising because, in traditional pipelines, forced alignment is used as an oracle alignment and normally leads to better perceptual quality than that of the predicted case. Since Tacotron learns both spectrograms and alignments simultaneously, it seems to produce the best spectrograms when it infers both of them. Among the baseline pipeline systems, as expected, a forced alignment gave higher scores than predicted alignment for both systems using vocoder parameters and mel-spectrogram. In the case of predicted alignment, the score has a long tail variance towards the low score region.

**Comparison of pipeline and Tacotron systems:** The best proposed system still does not match the quality of the best pipeline system. SA-Tacotron with accentual-type labels and the pipeline system using mel-spectrogram and predicted alignment had 3.60 ± 0.03 and 3.90 ± 0.03, respectively. These are not the same results as for the English experiments reported in [3]. One major difference of our proposed systems from pipeline systems other than architecture is input linguistic features; our proposed systems use phoneme and accentual-type labels only, but the baseline pipeline systems use various linguistic labels including word-level information such as inflected forms, conjugation types, and part-of-speech tags. In particular, an investigation on the same Japanese corpus found that the conjugation type of the next word is quite useful for \( F_0 \) prediction [20].

4. CONCLUSION

In this paper, we applied Tacotron to Japanese to extend it to a pitch-accent language. We proposed phone-based Tacotrons with and without accentual-type labels, one with self-attention layers to capture long term information better, and one using vocoder parameters including fundamental frequency. We conducted objective and subjective evaluations. Among the proposed systems, Tacotron with the self-attention extension outperformed that without self-attention both with and without labels. However, we revealed that, unlike experiments reported for English, the quality of traditional pipeline systems is better than the proposed systems for Japanese. We also found that choosing vocoder parameters is beneficial to pipeline systems, but this is completely opposite for the case of Tacotron.

One major difference of our proposed systems from the pipeline systems is the absence of word level information in linguistic features, so incorporating this information may improve the quality of the proposed systems and bring them up to the pipeline system’s level. Our next step towards end-to-end speech synthesis in various languages is to incorporate word-level information such as Kanji.

Acknowledgements We are grateful to Prof. Zhen-Hua Ling from USTC for kindly answering our questions.
5. REFERENCES


