Investigating accuracy of pitch-accent annotations in neural network-based speech synthesis and denoising effects

Hieu-Thi Luong¹, Xin Wang¹, Junichi Yamagishi¹, Nobuyuki Nishizawa²

¹National Institute of Informatics, Tokyo, Japan ²KDDI Research Inc., Saitama, Japan
{lhuonghieuthi,wangxin,jyyamagis}@nii.ac.jp, no-nishizawa@kddi-research.jp

Abstract
We investigated the impact of noisy linguistic features on the performance of a Japanese speech synthesis system based on neural network that uses WaveNet vocoder. We compared an ideal system that uses manually corrected linguistic features including phoneme and prosodic annotation in training and test sets against a few other systems that use corrupted linguistic features. Both subjective and objective results demonstrate that corrupted linguistic features, especially those in the test set, affected the ideal system’s performance significantly in a statistical sense due to a mismatch condition between the training and test sets. Interestingly, while an utterance-level Turing test showed that listeners had a difficult time differentiating synthetic speech from natural speech, it further indicated that adding noise to the linguistic features in the training set can partially reduce the effect of the mismatch, regularize the model, and help the system perform better when linguistic features of the test set are noisy.

Index Terms: speech synthesis, deep neural network, Japanese prosody, WaveNet

1. Introduction
Because of the rapid development of deep learning, more and more text-to-speech (TTS) synthesis systems adopt end-to-end approaches to some degree [1, 2, 3]. Although it has been reported that neural-style synthetic speech from one system achieved a similar degree of quality and naturalness to natural recordings [3], it is unknown how the end-to-end approach could perfectly avoid incorrect pronunciation [3] and make it possible to control prosody like the conventional structured architectures [4, 5, 6, 7]. More importantly, since most of the existing commercial TTS systems still adopt the pipeline structure which contains a front-end and a back-end, rapid shifting to an end-to-end architecture may be unable to answer how each part of the conventional structure contributes and limits the performance of existing TTS systems. Therefore, we believe that investigation on the pipeline of conventional TTS systems is still necessary and meaningful. In this work we adopted the conventional speech synthesis architecture which consists of three separate components: a linguistic analyzer, a neural network-based acoustic model [8, 9] and a vocoder to synthesize waveforms from acoustic features.

As the initial step, our previous work [10] has showed that the conventional TTS pipeline can be improved by replacing a deterministic vocoder [11, 12] and RNN-based acoustic models [13, 14] in the back-end with more advanced statistical models such as the WaveNet-based vocoder [15, 16]. However, our analysis revealed that the gap between synthetic speech and natural recordings still exists. One reason may be due to the fact that the statistical models in our previous work were trained by using linguistic feature automatically extracted from text. This motivated us to investigate the impact of the accuracy of the features on the back-end of the TTS system. There is a relevant investigation on the accuracy of phone sequences used for training of hidden Markov models [17]. Our main focus in this paper is the accuracy of pitch accent information and a neural network.

In this study, we first built an oracle system where manually corrected linguistic features were used for both model training and testing. Then, we compared the performance of the system with a few other systems that used corrupted linguistic features at training or/and testing stages. More particularly, we corrupted Japanese pitch accent types by adding discrete noise. From large-scale crowdsourcing listening tests, we found that in our neural network-based speech synthesis system, using corrupted linguistic features has a regularization effect (like a denoising auto-encoder) when linguistic features in the test set are noisy. We believe that this is a new finding in the speech synthesis field.

In section 2 and 3, we describe the statistical models and linguistic features used in our TTS systems. In section 4, we explain the methodology used to train and test our systems by using linguistic features with a varied amount of noise. In section 5, we list the results of both objective and subjective evaluation. Finally, in section 6, we discuss the findings and draw a conclusion.

2. Speech synthesis back-end
The back-end of the TTS system we investigated consists of two parts. The first part contains acoustic models that convert linguistic features into acoustic features such as the mel-generalized cepstral coefficients (MGC) and quantized fundamental frequency (F0). The second part is a WaveNet vocoder that generates speech waveform based on the basic of acoustic features. All of the models adopt the configurations used in our previous work [10].

2.1. Acoustic models
The acoustic models are trained to learn the mapping from a sequence of linguistic features \(l_1, ..., l_N\) to a sequence of acoustic features \(a_1, ..., a_N\) where \(N\) denotes the total number of frame. While a vanilla neural network can be used as the acoustic model, it assumes that \(\{a_1, a_2, ..., a_N\}\) is a set of independent random variables given \(l_1, ..., l_N\) even if convolution or recurrent layers are used. To overcome such weakness, we used autoregressive models, the basic idea of which is to feed the target data of the previous step as the input of the current step. On the basis of this idea, two separate autoregressive models plotted in Figure 1 were trained to model MGC and quantized F0, respectively.

The model for MGC was referred to as shallow autoregres-
of recurrent network (SAR). SAR maps a sequence of linguistics features to the value of a parameter set of which the distribution (in this case, a Gaussian distribution) of MGC of each frame can be specified. Different from a normal mixture density network [18], SAR uses a linear function to summarize the acoustic features in previous frames and then changes the distribution of the current frame. A similar network was used for quantized F0, which is referred to as deep autoregressive recurrent current network (DAR). DAR was trained to map linguistic features to a quantized F0 representation rather than interpolated continuous-valued F0 data. Another distinct feature of DAR in comparison with SAR is that the output of the network is fed back to a recurrent layer that is closer to the input side.

The structure of the acoustic modelling networks are illustrated in Figure 1. Bi-directional and uni-directional long-short term memory (LSTM) layers were used after feedforward layers. Details on these models are given in our previous papers [19, 14].

### 2.2. WaveNet vocoder

To improve the quality of synthetic speech, we used a speaker dependent WaveNet vocoder. The WaveNet vocoder is a CNN-based autoregressive network that models a conditional distribution of a waveform sequence $o_{1:T}$ over an auxiliary feature sequence $a_{1:N}$ as

$$p(o_{1:T}|a_{1:N}) = \prod_{t=1}^{T} p(o_t|o_{<t}, a_{1:N}).$$  \hspace{1cm} (1)

For each sample $o_t$ at time $t$, its value is conditioned on all of the previous observations $o_{<t}$. In practice, the prediction of $o_t$ was limited to a finite number of previous samples, which together were referred to as receptive field. By sequentially sampling the waveform per time step, the WaveNet vocoder can produce very high-quality synthetic speech in terms of naturalness, as reported in several papers [15, 3, 16].

### 3. Linguistic features used for our Japanese TTS system

The linguistic features used for conventional Japanese TTS systems mainly include segmental and supra-segmental linguistic information. Despite the numerous differences in the two sets of linguistic features used in the experiments, i.e., OpenTalk and oracle that will be introduced in Section 4.1, both sets contain quinphone contexts, word part-of-speech tags, pitch accent types of the accent phrases, interrogative phrase marks, and other structural information such as the position of the mora in a word, accent phrases, and utterances. These linguistic features will be used as the input of the acoustic model.

The two types of linguistic features that we were interested in for this investigation include the pitch accent type (Acc_Type) and the interrogative phrase mark (Question_Flag). The value of the pitch accent type is equal to the location of the accented mora in a Japanese accent phrase. It can also be a special number such as 0, which indicates a no-accent phrase. The interrogative phrase mark is binary and indicates whether a phrase is interrogative or not. These two types of features are essential to the prosody of Japanese utterances yet difficult to accurately obtain by using automatic prosodic annotation or text-analysis.

### 4. Experiments

#### 4.1. Data and features

This study used the same speech corpus as our previous work [10]. This corpus has high-quality speech recordings of a female voice talent and was released as part of the Ximera datasets [20]. Compared with our previous work, we excluded hundreds of utterances in which the manually annotated labels were unusable due to imperfect pronunciation. This new training set contained 27,999 utterances while both the validation and test set contained 480 utterances. The duration of the training set was about 46,914 hours, among which the total silence at the two ends of the utterances was around 13.393 hours in total. The duration of the validation and test sets was 0.815 and 0.824 hours.

Acoustic features were extracted by using WORLD [12] spectral analysis modules and SPTK. We used speech waveforms at a sampling frequency of 48 kHz to obtain these features with a window length of 25 ms and frame shift of 5 ms. 60-dimensional Mel-generalized cepstral coefficients (MGCs) and 25-dimensional band-limited aperiodicity values (BAP) were extracted. F0s were quantized into 255 levels as described in [14]. To investigate the impact of the accuracy of linguistic features, we prepared three sets of linguistic features:

- **OpenTalk**: the first set of linguistic features was extracted automatically from text by using OpenTalk [21]. These features were converted into 389 dimensional vector. This set is included as a reference because it was used in our previous work.
- **Oracle**: the second set of linguistic features is based on in-house annotations provided by KDDI Research, Inc. The definition of the linguistic features is very similar to that used in the above first set, but it contains more precise phone definitions. Part-of-speech tagging is not included in the annotations. The dimension of the linguistic feature vector was 265. All features were manually verified.
- **Corrupted**: the third set is based on the second set. However, we randomly changed the values of certain linguistic features. More specifically, we randomly added discrete noise ranging between -2 and +2 to the original value of [Acc_Type] for each accent phrase with a 50% probability. The value of the binary feature [Question_Flag] for each accent phase was also randomly converted to the opposite value with a 30% probability. We expected that these two types of processing would reproduce the annotation errors of Japanese accent types and
Table 1: Definition and notations of TTS systems used in this experiment. Definition of linguistic feature sets is given in Section 4.1. Oracle + Corrupted means that 7999 randomly selected training utterances used corrupted labels, while rest used oracle linguistic features. Also shown are RMSE values and correlation of generated F0 and U/UV errors compared with natural speech. Mel-cepstral distortion (MCD) between generated and natural MGCs is also shown.

<table>
<thead>
<tr>
<th>Notations</th>
<th>Train set</th>
<th>Test set</th>
<th>RMSE</th>
<th>CORR</th>
<th>V/U Error</th>
<th>MCD [dB]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Linguistic features</td>
<td>Objective measures</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MOO</td>
<td>Oracle</td>
<td>Oracle</td>
<td>29.44</td>
<td>0.90</td>
<td>3.40%</td>
<td>4.72</td>
</tr>
<tr>
<td>MOC</td>
<td>Oracle</td>
<td>Corrupted</td>
<td>23.31</td>
<td>0.94</td>
<td>3.25%</td>
<td>4.63</td>
</tr>
<tr>
<td>MMC</td>
<td>Oracle + Corrupted</td>
<td>Corrupted</td>
<td>31.09</td>
<td>0.89</td>
<td>3.29%</td>
<td>4.66</td>
</tr>
<tr>
<td>MOC</td>
<td>Oracle + Corrupted</td>
<td>Oracle</td>
<td>28.15</td>
<td>0.91</td>
<td>3.20%</td>
<td>4.64</td>
</tr>
<tr>
<td>MMO</td>
<td>Oracle + Corrupted</td>
<td>Oracle</td>
<td>24.44</td>
<td>0.94</td>
<td>3.15%</td>
<td>4.63</td>
</tr>
</tbody>
</table>

4.2. Model configurations

The structure of the acoustic models is plotted in Figure 1. The configuration of the layer size was 512 for feedforward, 256 for bi-directional LSTM-RNN, and 128 for uni-directional LSTM-RNN. The size of a linear layer depends on the size of the output. For SAR network, the output is a parameter set of Gaussian distributions for MGC, BAP, and voiced/unvoiced (V/UV) flags. BAP and V/UV were also included in the output even though they are not used to generate speech waveforms with the WaveNet vocoder. DAR used a similar configuration of layer size as SAR, but the output layer was a hierarchical softmax layer.

Although the acoustic features were extracted from speech waveforms at a sampling frequency of 48 kHz, the WaveNet vocoder was trained by using speech waveforms at a sampling frequency of 16 kHz. PCM waveform samples were quantized into 10 bits after they were compressed by $\mu$-law coding [22]. The network contained 40 causal dilated convolution layers similar to [23]. WaveNet blocks were conditioned on MGC and quantized F0 parameters locally. The WaveNet vocoder was trained on acoustic features extracted from natural speech, while, in the generation stage, MGC and quantized F0 features predicted from DAR and SAR models were used.

4.3. Experimental conditions

To investigate the impact of the noise in linguistic features, we trained a few systems by using different sets of linguistic features in the training and test stages as we described earlier. The definition and notations of each system can be found on the left part of Table 1. Note that the linguistic features for the validation set were not corrupted. Also note that, instead of using the results of our previous study, we retrained the OpenJTalk-based model (OJT) by using the same data set configuration described in Section 4.1. Thus, the results of OJT can be compared with those of other experimental models.

5. Evaluations

5.1. Objective evaluation

Table 1 shows the performance of each system in terms of RMSE, correlation, and V/UV errors of F0 trajectories converted from predicted F0 classes including an unvoiced class. As expected, the model trained and tested by using manually annotated labels, i.e., MOO, achieved the best results among the systems for all of the measurements. We can also see that when testing on the corrupted linguistic labels, the performance of MOC drastically dropped.

Interestingly, MMC and MMO, which were trained by using partially corrupted linguistic features, performed better than MOC. For MMO, the objective results are comparable to those of MOO, which suggests that 7999 (around 28.57% of all training data) corrupted labels did not affect the overall quality significantly. Meanwhile, MMC performed better than MOC even though MMC used corrupted linguistic features for training.

Our hypothesis is that mixing corrupted labels with the training data is similar to a regularization method, such as the denoising auto-encoder and doing so helps a model generalize better and eases the negative impacts of the wrong information provided from incorrect linguistic features in the testing stage.

5.2. Subjective evaluation

The objective evaluation hinted at the performance of the acoustic models. However, because we used WaveNet vocoder and its sampling rather than a traditional deterministic vocoder to synthesize speech waveforms, it was necessary to test the overall quality of the synthetic speech samples subjectively.

Therefore, we also conducted a large-scale subjective test. Synthetic speech samples were generated by using the WaveNet vocoder. The natural speech was downsampled to 16 kHz and further converted to 8-bit $\mu$-law. Synthetic speech was converted to 8-bit and was normalized to have a similar volume to natural speech using the sv56 program [24].

With the above five systems and the natural speech (NAT), each of which contained 480 utterances from the test set, we conducted two subjective tests. The first test was done to evaluate the mean-opinion-score (MOS) on a five point scale. The second was similar to a Turing test, where participants were asked to identify which of two samples presented is synthetic. During this test, an anchor question was included, where we presented the same natural speech twice. This question was expected to provide some insight into the nature of the testing environment. No default answers were given in any of these tests to make sure participants would have to make their own choices.

This large-scale listening test was conducted online through crowd-sourcing. Each participant was asked to navigate twelve pages for each set. Each page contained two questions, one for the MOS and another for the Turing test. The audio sample for the quality question contained a different sentence from that for the Turing question on the same page. One hundred subjects participated. They were allowed to repeat the test up to ten times. We collected a total of 720 sets, which led to 3 data points per unique audio sample for all of the systems.

Quality test: Figure 2 shows subjective results for the quality test with a 95% confidence interval with a student’s t-distribution. Unsurprisingly natural speech still achieved the highest and most statistically significant score at 3.96 even when converted to the $\mu$-law encoding format. Audio samples generated by using manually annotated labels at the generation
stage (MOO and MMO) achieved the second highest score, and
the difference between MOO and MMO was not statistically
significant (3.62 versus 3.63, $p$-value=0.720). We can also see
that OJT and MOC performed the worst, and the difference be-
tween them was not statistically significant (3.33 versus 3.26,
$p$-value=0.05). Note that the $p$-value was calculated with Holm-
Bonferroni correction.

What’s interesting is that MMC, which used the corrupted
linguistic features at both the training and testing stages, was
better than OJT and MOC. These subjective results were con-
sistent with the results of the objective evaluation on F0.

These results indicate a correlation between the accuracy of
the linguistic features and the quality of the synthetic speech. A
greater impact could be seen if the accuracy of annotated labels
is high at the testing stage instead of the training stage. Another
finding is that, when linguistic features used in the test set con-
tained noises, training the neural network models with a small
amount of corrupted linguistic features seemed to improve
the quality of the synthetic speech. We can also see that adding
a small amount of corrupted linguistic features in the training set
did not degrade the quality of the synthetic speech even if the
test set did not contain any noise.

Turing (Identification) test: for the Turing test, partici-
pants were asked to identify which of two audio samples pre-

tained on left or right side of the web page was synthetic. The
audio samples from one of the TTS systems and from natu-
ral speech were randomly switched between left and right to
discourage subjects from developing any bias patterns. Fig-
ure 3 shows the result. Surprisingly, for all comparisons be-
tween the synthetic and natural speech utterances, the correct-
identification ratio was around 50%, which suggested that our
participants could not decide with certainty which of the two
samples presented was synthetic. There was no significant dif-
ference between the five pairs of generated and (slightly de-
graded) natural speech. We think that this may not be surprising
because the correlation of our F0 prediction model was as high
as 0.9, we used a very large speaker-dependent corpus, that was
larger than in a recent paper on Google’s Tacotron 2 [3], and
natural speech was also slightly degraded by the $\mu$-law coding.

As we included an anchor test in our evaluation in which
participants were asked to judge the differences between the
same natural speech, it may be helpful to look into the result
to gain some insight into our testing environment. The results
we got for this anchor test showed that the left options were fa-
vored 60% of the time, which suggested that participants had a
slight bias for left option when it was difficult to choose the cor-
rect one. We can also analyze whether participants had a slight
bias for left options for comparisons between the synthetic and
natural speech utterances. Although the two options were ran-
domly switched, from two sub figures at the bottom, we can see
that the same tendency exists regardless of system types used.
This basically gives some insight into developing a more so-
phisticated Turing test in the future.

With the outcomes for the Turing test, we can conclude that,
while the synthetic speech did not achieve the same quality as
natural speech, it was difficult for a normal human being to cor-
correctly determine the synthetic speech with our current state-of-
the-art setups, at least when a reference natural-speech utterance
was not offered.

6. Conclusions

In this paper, we investigated the impact of noisy linguistic fea-
tures on the performance of a Japanese speech synthesis sys-
tem based on neural network that uses WaveNet vocoder. In
this investigation, an ideal system that used manually corrected
linguistic features in the training and test sets was compared
against a few other systems that used corrupted linguistic fea-
tures. The corrupted linguistic features, which were created by
adding noises artificially to the correct pitch accent information.

Both subjective and objective results demonstrate that cor-
rupted linguistic features, especially those in the test set, af-
ected our TTS system’s performance significantly in a statis-
tical sense due to mismatched conditions between the training
and test sets. It was further indicated that adding noise to the lin-
guistic features in the training set can partially reduce the effect
of the mismatch, regularize the model, and help the system per-
form better when the linguistic features of the test set are noisy.

As far as we know, this is a new finding in the speech synthesis
field. Interestingly the utterance-level Turing test showed that
our listeners had a difficult time differentiating synthetic speech
from slightly degraded natural speech.

Our future work includes comparing of our TTS system us-
ing manually corrected labels with recent end-to-end TTS sys-
tems and evaluating without using $\mu$-law coding.
References


