The NITech text-to-speech system for the Blizzard Challenge 2017

Kei Sawada, Kei Hashimoto, Keiichiro Oura, Keiichi Tokuda

Nagoya Institute of Technology, Nagoya, JAPAN
{swdkei, bonanza, uratec, tokuda}@sp.nitech.ac.jp

Abstract

This paper describes a text-to-speech (TTS) system developed at the Nagoya Institute of Technology (NITech) for the Blizzard Challenge 2017. In the challenge, about seven hours of highly expressive speech data from children’s audiobooks were provided as training data. For this challenge, we redesigned linguistic features for statistical parametric speech synthesis based on audiobooks. Furthermore, we introduced the parameter trajectory generation process considering the global variance into the training of mixture density network based acoustic models. Large-scale subjective evaluation results show that the NITech TTS system achieved naturally sounding and intelligible synthesized speech.

Index Terms: text-to-speech system, statistical parametric speech synthesis, deep neural network, Blizzard Challenge, audiobook

1. Introduction

A number of studies on text-to-speech (TTS) systems have been conducted. Consequently, the quality of synthetic speech has improved, and such systems are now used in various applications, such as for in-car navigation, smartphones, and spoken dialogue systems. Accordingly, the demand for TTS systems offering high-quality synthetic speech, various speaking styles, and various languages is increasing.

Although many TTS systems have been proposed, comparisons of such systems are difficult when the corpus, task, and listening test are different. The Blizzard Challenge was started in order to better understand and compare research techniques in constructing corpus-based speech synthesizers with the same data in 2005 [1]. This challenge has so far provided English, Mandarin, some Indian languages, English audiobooks, etc. as training data. The series of Blizzard Challenges has helped us measure progress in TTS technology [2].

As computer processing power increased, approaches based on big data have been successful in various research fields. In corpus-based speech synthesis, a quality of synthesized speech was improved by using a large amount of training data. Therefore, a TTS system based on big data is important in speech synthesis research. Speech data recorded with less noise and under the same recording conditions are suitable for training TTS systems. A large amount of training data is also necessary to synthesize various speaking styles. For this reason, recording a large amount of speech data for a TTS system requires a huge cost. Therefore, TTS system construction method based on audiobooks has received considerable attention. Audiobooks can be relatively easily collected as a large amount of speech data and text pairs. In the Blizzard Challenge 2013, around 300 hours of audiobooks were provided as training data [3]. In the Blizzard Challenge 2016, about five hours of highly expressive speech data from professionally produced English children’s audiobooks were provided [4]. In the Blizzard Challenge 2017, about seven hours of speech data from children’s audiobooks, which includes the five hours released in the Blizzard Challenge 2016, were provided as training data [5]. All 56 books were recorded by one native British English female professional speaker. Texts corresponding to speech data were also provided. The task was to construct a speech from this data that is suitable for reading audiobooks to children.

The Blizzard Challenge has been submitting statistical parametric speech synthesis (SPSS) systems to the Blizzard Challenge since 2005. Typical SPSS systems have three main components: linguistic features estimation, acoustic features estimation, and speech waveform generation. In the linguistic features estimation component, linguistic features, e.g., phonemes, syllables, accents, and parts-of-speech, of an input text is estimated. In the acoustic features estimation component, acoustic features which express characteristics of a speech waveform is estimated with the linguistic features. In the speech waveform generation component, a speech waveform is generated from the acoustic features.

The rest of this paper is organized as follows. Section 2 describes the NITech TTS system for the Blizzard Challenge 2017. Subjective listening test results are given in Section 3, and concluding remarks and an outline for future work are presented in the final section.

2. NITech TTS system

The provided audiobooks contained mismatches between speech data and text. These mismatches were caused by the misreading of a text or words that do not exist in the text, i.e., description of a book or onomatopoeia. This will negatively affect training of the statistical parametric speech synthesis (SPSS). To overcome this problem in last year’s challenge, we investigated the automatic construction of a training corpus from audiobooks using a speech recognizer. Figure 1 shows an overview of the automatic training corpus construction method. The details were described in [6].

Figure 2 gives an overview of the Nagoya Institute of Technology (NITech) text-to-speech (TTS) system for the Blizzard Challenge 2017. In the training part, linguistic and acoustic features are first extracted from text analysis and vocoder encoding, respectively. Second, hidden Markov model (HMM)-based
To consider sentence structures, linguistic features of sentence-level syntactic and dependency parsing are performed. The results of parsing are represented by tree structures, which are called syntactic tree and dependency tree, respectively. Information obtained from the trees is used as linguistic features.

Children’s audiobooks include various speaking styles. Especially, the speech data in the conversational part and exclamatory sentences of audiobooks are read emphatically, emotionally, and so on. These speaking styles in speech data should be distinguished by linguistic features. For this reason, linguistic features based on double quotes and types of sentences are used to express the reading styles of speech data.

The training corpus contains various speaking variations for each phrase. In order to train high-quality acoustic model, it is necessary to distinguish speaking variations. We introduce a phrase code into the linguistic features. The phrase code is a unique value assigned to each phrase in the training corpus. The phrase code is able to distinguish speaking variations between phrases in model training. In the synthesis part, a speaking style can be represented by using an appropriate phrase code. However, it is costly to select a phrase code for each test phrase. Therefore, we investigate a framework to automatically select an appropriate phrase code of a test phrase from phrase codes in the training corpus. The doc2vec [13] which is a technique of document vectorization proposed in the field of natural language processing is used. It becomes possible to measure phrase similarity by vectorizing phrases. The phrase code of the highest similarity phrase in the training corpus is used the phrase code of the test phrase. For example, when a test phrase is an angry phrase, the degree of similarity between the test phrase and the angry phrase is high.

A synthesized speech is generated from vocoder decoding. To synthesize high-quality speech, we used a mixture of DNN training and DNN-based spectral and excitation models. In SPSS using DNN-based acoustic models [15], a single DNN is trained to represent a mapping function from linguistic features to acoustic features. In the synthesis part, the linguistic features extracted from given text to be synthesized are mapped to acoustic features by using the trained DNN using forward-propagation. To synthesize high-quality speech, we used a mixture density network (MDN) as an acoustic model [10][11] and applied trajectory training considering global variance (GV) [9].

2.2. DNN-based SPSS

In SPSS using DNN-based acoustic models [15], a single DNN is trained to represent a mapping function from linguistic features to acoustic features. In the synthesis part, the linguistic features extracted from given text to be synthesized are mapped to acoustic features by using the trained DNN using forward-propagation. To synthesize high-quality speech, we used a mixture density network (MDN) as an acoustic model [10][11] and applied trajectory training considering global variance (GV) [9].

2.2.1. MDN-based SPSS

A speech parameter vector $o_t$ consists of a $D$-dimensional static-feature vector $c_t = [c_1(1), \ldots, c_1(D)]^\top$ and both of its first- and second-order dynamic feature vectors, $\Delta^{(1)}c_t$ and $\Delta^{(2)}c_t$.

$$o_t = [c_t^\top, \Delta^{(1)}c_t^\top, \Delta^{(2)}c_t^\top]^\top$$  \hspace{1cm} (1)

The sequences of speech parameter vectors $o$ and static-feature vectors $c$, which represent a page in our system, can be written in vector forms as follows:

$$o = [o_1^\top, \ldots, o_T^\top]^\top$$  \hspace{1cm} (2)

$$c = [c_1^\top, \ldots, c_T^\top]^\top$$  \hspace{1cm} (3)

Speech synthesizer [12] is constructed to estimate phoneme-level alignments. Finally, deep neural network (DNN)-based speech synthesizer is constructed by using frame-by-frame linguistic and acoustic features. In the synthesis part, acoustic features are estimated from linguistic features using the HMM-based duration model and DNN-based spectral and excitation models. A synthesized speech is then generated from vocoder decoding. The details of linguistic features for audiobooks and DNN-based SPSS are described in the following sections.

Figure 1: Overview of training corpus construction (SI: speaker independent, SA: speaker adapted, BA: book adapted, SR: speech recognizer, AM: acoustic model, LM: language model)

Figure 2: Overview of the NUtech TTS system
where \( T \) is the number of frames included on a page. The relation between \( o \) and \( c \) can be represented as \( o = Wc \), where \( W \) is a window matrix extending \( c \) to \( o \). The optimal static-feature vector sequence is obtained by

\[
\hat{c} = \arg \max_{\mathbf{c}} P(o | \lambda) = \arg \max_{\mathbf{c}} \mathcal{N}(\mathbf{Wc} | \mu, \Sigma) \tag{4}
\]

where \( \lambda \) is a parameter set and \( \mathcal{N}(\cdot | \mu, \Sigma) \) denotes the Gaussian distribution with a mean vector \( \mu \) and covariance matrix \( \Sigma \). The optimal static-feature sequence \( \hat{c} \) is given by

\[
\hat{c} = PW^T \Sigma^{-1} \mu, \quad P = (W^T \Sigma^{-1} W)^{-1} \tag{5}
\]

As a result, smooth static-feature trajectories can be obtained using dynamic features as constraints.

An MDN maps a linguistic-feature vector \( l \) to parameters of a Gaussian mixture model (GMM). In this challenge, we used a single MDN as an acoustic model. Assuming that outputs of a neural network are used as mean and standard deviation parameters in a statistical model, an objective function can be defined as

\[
\mathcal{L} = P(o | l, \lambda_{MDN}) = \mathcal{N}(o | \mu, \Sigma) = \prod_{t=1}^{T} \mathcal{N}(o_t | \mu_t, \Sigma_t) \tag{6}
\]

where the mean and covariance parameter are obtained by \( \mu_t = [\mu_{t1}, \mu_{t2}, \ldots, \mu_{td}] \) and \( \Sigma_t = \text{diag}[\sigma_{t1}^2, \sigma_{t2}^2, \ldots, \sigma_{td}^2] \), respectively. Then, the mean and standard deviation at frame \( t, \mu_{t,d} \) and \( \sigma_{t,d} \), can be obtained as follows:

\[
\mu_{t,d} = g_{d}^{(\mu)}(l_t, \lambda_{MDN}) \tag{7}
\]

\[
\sigma_{t,d} = \exp(g_{d}^{(\sigma)}(l_t, \lambda_{MDN})) \tag{8}
\]

where \( g_{d}^{(\mu)}(l_t, \lambda_{MDN}) \) and \( g_{d}^{(\sigma)}(l_t, \lambda_{MDN}) \) are the activations of the output layer corresponding to mean and standard deviation parameters, given \( l_t \) and \( \lambda_{MDN} \), respectively. The MDN parameter set \( \lambda_{MDN} \) is optimized in the sense of maximum likelihood as follows:

\[
\lambda_{MDN} = \arg \max_{\lambda_{MDN}} P(o | l, \lambda_{MDN}) \tag{9}
\]

The MDN can be trained by standard back-propagation.

2.2.2. Trajectory training

In the MDN-based SPSS framework, although the frame-level objective function is used for training a MDN, the sequence-level (page-level) objective function is used for parameter generation. To address this inconsistency between training and synthesis, a trajectory training method is introduced into the training process of MDNs.

The traditional likelihood function in Eq. (6) can be reformulated as a trajectory likelihood function by imposing the explicit relationship between static and dynamic features, which is given by \( o = Wc \) [16]. The trajectory likelihood function of \( c \) is then written as

\[
\mathcal{L}_{T} = \frac{1}{Z} P(o | l, \lambda) = P(c | l, \lambda) = \mathcal{N}(c | \hat{c}, P) \tag{10}
\]

where \( Z \) is a normalization term. Inter-frame correlation is modeled by the covariance matrix \( P \) that is generally full. Note that the mean vector \( \hat{c} \) is equivalent to the generated static-
feature sequence expressed by Eq. (5). The parameter set $\lambda$ is estimated by maximizing the trajectory likelihood $L_{\text{Tj}}$.

2.2.3. Trajectory training considering GV

To address the over-smoothing problem of generated parameter trajectories, the concept of parameter generation considering the GV was introduced into the training of DNNs [9]. In this challenge, we introduce the trajectory training considering the GV into the training of a MDN-based acoustic model. Figure 3 shows an overview of trajectory training considering the GV. The objective function $L_{\text{GVTj}}$ is given by

$$L_{\text{GVTj}} = P(c \mid l, \lambda)P(v(c) \mid l, \lambda, \lambda_{\text{GV}})^{wT} = N(c \mid \bar{c}, P)N(v(c) \mid v(\bar{c}), \Sigma_{\text{GV}})^{wT}$$

where $v(c) = [v(l), \ldots, v(D)]^T$ is a GV vector of the static-feature vector sequence $c$. The GV vector is calculated page by page as follows:

$$v(d) = \frac{1}{T} \sum_{t=1}^{T} (c_t(d) - \langle c(d) \rangle)^2, \quad \langle c(d) \rangle = \frac{1}{T} \sum_{t=1}^{T} c_t(d)$$

where $d$ is an index of the feature dimension. The mean vector of the probability density for the GV, $v(\bar{c})$, is defined as the GV of the mean vector of the trajectory likelihood function in Eq. (6), which is equivalent to the GV of the generated parameters expressed by Eq. (7). The GV likelihood $P(v(c) \mid l, \lambda, \lambda_{\text{GV}})$ works as a penalty term to make the GV of the generated parameters close to that of the natural ones. The balance between the two likelihoods $P(c \mid l, \lambda)$ and $P(v(c) \mid l, \lambda, \lambda_{\text{GV}})$ is controlled by the GV weight $w$. The parameter set $\lambda$ which consists of the parameter of the MDN $\lambda_{\text{MDN}}$ and the covariance matrix $\Sigma_{\text{GV}}$ of the GV vector, is estimated by maximizing the objective function $L_{\text{GVTj}}$. The parameters are optimized so that the GVs of generated trajectories get close to the natural ones.

The optimal static-feature vector sequence $\hat{c}$ is determined by maximizing the objective function $L_{\text{GVTj}}$ as follows:

$$\hat{c} = \arg \max \frac{1}{T} \sum_{t=1}^{T} (c_t(d) - \langle c(d) \rangle)^2$$

Since this estimate is equivalent to the maximum likelihood estimate by using the basic parameter generation algorithm expressed by Eq. (8), the basic parameter generation algorithm can be used for this framework.

3. Blizzard Challenge 2017 evaluation

3.1. Training corpus construction conditions

The collection of provided children’s audiobooks consisted of 56 books with a total 1258 pages. An SR was trained to construct a training corpus for SPSS. The CMU Pronouncing Dictionary [17] and the WSJ0, WSJ1 [18], and TIMIT [19] databases were used to train the SR. Speech signals were sampled at a rate of 16 kHz and windowed by a 25-ms hamming window with a 10-ms shift. The acoustic-feature vector consisted of 39 components composed of 12-dimensional mel-frequency cepstral coefficients (MFCCs) including the energy with the first- and second-order derivatives. A three-state left-to-right GMM-HMM without skip transitions was used. The trained GMMs had 32 mixtures for pause and 16 mixtures for the other phonemes. A tri-gram LM was created based on the text of the provided children’s audiobooks. The HTK [20] and SRILM [21] were used to construct the SR. The training recipe was the same as that of the HTK Wall Street Journal Training Recipe [22]. Thresholds of word-match accuracy for adaptation and training corpora were set to 90% [8]. After pruning, the training corpus for SPSS consisted of 921 pages.

3.2. TTS system construction conditions

Linguistic features were extracted using Festival [23], Stanford Parser [24], SyntaxNet [25], and gensim [26]. The speech signals were sampled at a rate of 44.1 kHz and windowed with a fundamental frequency ($F_0$)-adaptive Gaussian window with a 5-ms shift. Votting results concerning $F_0$ (estimated by using RAPT [27], SWIPE’ [28], and REAPER [29]) were taken as $F_0$ of acoustic features.

The HMM-based SPSS system was constructed to estimate phoneme-level alignments. The acoustic-feature vectors were composed of 228 dimensions: 49-dimension STRAIGHT mel-cepstral coefficients including the 0th coefficient, $F_0$, 24-dimension mel-cepstral analysis aperiodicity measures, and their first- and second-order derivatives. A five-state left-to-right context-dependent multi-stream multi-space probability distribution hidden semi-Markov model (MSD-HSMM) [31, 32, 33, 34] without skip transitions was used as the acoustic model. Each state output probability distribution was composed of a spectrum, $F_0$, and aperiodicity streams. The spectrum and aperiodicity streams were modeled using single multi-variante Gaussian distributions with diagonal covariance matrices. The $F_0$ stream was modeled using an MSD consisting of a Gaussian distribution for voiced frames and a discrete distribution for unvoiced frames. State durations were modeled using a Gaussian distribution. The HTS [35] and SPTK [36] were used for constructing the HMM-based SPSS system.

In the MDN-based SPSS system, the input feature was a 1685-dimensional feature vector consisting of 925 linguistic features including binary features and numerical features for contexts, 10 duration features, 150-dimensional word code, and 600-dimensional phrase code. Fix-dimensional normally distributed random vector was used as word and phrase codes, and pre-trained word2vec and doc2vec were used to measure word and phrase similarity. The output feature was a 107-dimensional feature vector consisting of 69-dimension STRAIGHT mel-cepstral coefficients, $F_0$ acquired by linearly interpolating val-
ues in unvoiced parts, voiced/unvoiced binary value, and 34-dimension mel-cepstral analysis aperiodicity measures. The input features were normalized to be within 0.0–1.0 based on their minimum and maximum values in the training data, and the output features were normalized to have zero-mean variance. The input and output features were time-aligned frame-by-frame by using the trained MSD-HSMM. A single MDN, which models spectral, excitation, and aperiodicity parameters, was trained. The architecture of the MDNs was three hidden layers with 8000 units per layer. The sigmoid activation function was used in the hidden layers and the linear activation function was used in the output layer. For training the MDNs, a mini-batch stochastic gradient descent (SGD)-based optimization function was used in the output layer. For training the hidden layers with 8000 units per layer. The sigmoid activation function was used in the hidden layers and the linear activation function was used in the output layer. For training the MDNs, a mini-batch stochastic gradient descent (SGD)-based back-propagation algorithm and dropout with a probability of 0.6 were used. The GV weight \( w \) was set to 0.001 in Eq. (11). Dynamic range compressor (DRC) was applied to power of synthesized speech.

### 3.3. Experimental conditions of listening test

Large-scale subjective listening tests were conducted by the Blizzard Challenge 2017 organization. The listeners included paid participants, speech experts, and volunteers. The paid participants (native speakers of English) took the test in soundproof listening booths using high-quality headphones. The speech experts and volunteers included non-native speakers of English.

To evaluate the page domain of a children’s book, 7-page-domain-criteria 60-point mean opinion score (MOS) tests were conducted. The terms in the parentheses were used to label the points 10 for “bad” and 50 for “excellent” on the scale. Listeners listened to one whole page from a children’s book and chose a score from 1 to 60 based on the following 7-page-domain-criteria.

- overall impression (OI): “bad” to “excellent”
- pleasantness (PL): “very unpleasant” to “very pleasant”
- speech pauses (SP): “speech pauses confusing/ unpleasant” to “speech pauses appropriate/pleasant”
- stress (ST): “stress unnatural/confusing” to “stress natural”
- intonation (IN): “melody did not fit the sentence type” to “melody fitted the sentence type”
- emotion (EM): “no expression of emotions” to “authentic expression of emotions”
- listening effort (LE): “very exhausting” to “very easy”

To evaluate the sentence domain of children’s book, 2-sentence-domain-criteria 5-point MOS tests were conducted. Listeners listened to one sample and chose a score from 1 to 5 based on the following 2-sentence-domain-criteria.

- naturalness (NAT): “completely natural” to “completely unnatural”
- similarity (SIM): “sounds like a totally different person” to “sounds like exactly the same person”

To evaluate intelligibility, the participants were asked to transcribe semantically unpredictable sentences (SUS) by typing in the sentence they heard. The average word error rate (WER) was calculated from these transcripts.

### 3.4. Experimental results

Table 2 lists the means and standard deviations of the listening test results from the all listeners. Systems A, B, C, D, and L represent the following systems.

- A: natural speech
- B: unit-selection benchmark system
- C: HMM benchmark system
- D: DNN benchmark system
- L: NITech system

The ordering of systems is in descending order of NAT. Wilcoxon’s signed rank tests were used to determine significance difference \( * \). In Table 2, asterisk * means a statistically significant difference between system L and other systems.

The page-domain results show that system L ranked 4th, 5th, 3th, 4th, 4th, and 3rd out of the 16 TTS systems listed in Table 2 for page-domain-criteria OI, PL, SP, ST, IN, EM, and LE, respectively. Only system L statistically significantly better than our system L except IN criterion. Overall, our system L achieved good performance. The sentence-domain results show that system L ranked 3rd and 7th for sentence-domain-criteria NAT and SIM, respectively. Our system L achieved naturally

<table>
<thead>
<tr>
<th>System</th>
<th>Page domain</th>
<th>Sentence domain</th>
<th>SUS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>OI</td>
<td>PL</td>
<td>SP</td>
</tr>
<tr>
<td>A</td>
<td>47 ± 9*</td>
<td>46 ± 10*</td>
<td>47 ± 9*</td>
</tr>
<tr>
<td>B</td>
<td>38 ± 10*</td>
<td>38 ± 10*</td>
<td>36 ± 11*</td>
</tr>
<tr>
<td>C</td>
<td>32 ± 10</td>
<td>31 ± 10</td>
<td>34 ± 11</td>
</tr>
<tr>
<td>D</td>
<td>31 ± 10</td>
<td>30 ± 10</td>
<td>31 ± 12</td>
</tr>
<tr>
<td>E</td>
<td>31 ± 12</td>
<td>32 ± 12</td>
<td>27 ± 13*</td>
</tr>
<tr>
<td>F</td>
<td>31 ± 11</td>
<td>31 ± 10</td>
<td>31 ± 11</td>
</tr>
<tr>
<td>G</td>
<td>27 ± 11*</td>
<td>27 ± 11*</td>
<td>25 ± 12*</td>
</tr>
<tr>
<td>H</td>
<td>26 ± 10*</td>
<td>25 ± 10*</td>
<td>24 ± 11*</td>
</tr>
<tr>
<td>I</td>
<td>26 ± 10*</td>
<td>27 ± 10*</td>
<td>25 ± 12*</td>
</tr>
<tr>
<td>J</td>
<td>25 ± 10*</td>
<td>26 ± 10*</td>
<td>23 ± 11*</td>
</tr>
<tr>
<td>K</td>
<td>25 ± 10*</td>
<td>23 ± 10*</td>
<td>30 ± 12</td>
</tr>
<tr>
<td>L</td>
<td>24 ± 9*</td>
<td>24 ± 9*</td>
<td>25 ± 11*</td>
</tr>
<tr>
<td>M</td>
<td>22 ± 9*</td>
<td>21 ± 9*</td>
<td>24 ± 11*</td>
</tr>
<tr>
<td>N</td>
<td>21 ± 10*</td>
<td>22 ± 10*</td>
<td>24 ± 11*</td>
</tr>
<tr>
<td>O</td>
<td>16 ± 8*</td>
<td>15 ± 8*</td>
<td>21 ± 11*</td>
</tr>
<tr>
<td>P</td>
<td>11 ± 7*</td>
<td>11 ± 7*</td>
<td>22 ± 12*</td>
</tr>
<tr>
<td>Q</td>
<td>8 ± 6*</td>
<td>8 ± 6*</td>
<td>16 ± 11*</td>
</tr>
</tbody>
</table>

Table 2: Evaluation results
sounding synthesized speech. By contrast, SIM was the average score compared with high score NAT. Up until now, as a weak point of SPSS systems, low speaker similarity been cited. Therefore, we should improve the speaker similarity by SPSS approaches. In terms of intelligibility, system L achieved the lowest WER.

4. Conclusion
We described the Nagoya Institute of Technology (NITech) text-to-speech (TTS) system for the Blizzard Challenge 2017. We redesigned linguistic features for statistical parametric speech synthesis (SPSS) based on audiobooks. Additionally, we introduced the parameter trajectory generation process considering the global variance into the training of mixture density network based acoustic models. Large-scale subjective evaluation results show that the NITech TTS system synthesized naturally sounding and intelligible speech. However, we need to improve speaker similarity by SPSS approaches. Future work includes improving robustness of outliers and introducing direct speech waveform prediction models, such as WaveNet [38], to avoid degradation of speech quality accompanying use of a vocoder.

5. Acknowledgements
This research and development work was partly supported by the MIC/SCOPE #162106106.

6. References