The NITech text-to-speech system for the Blizzard Challenge 2016

Kei Sawada, Chiaki Asai, Kei Hashimoto, Keiichiro Oura, and Keiichi Tokuda

Department of Scientific and Engineering Simulation,
Nagoya Institute of Technology, Nagoya, JAPAN
{swdkei, asai_cs, 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 2016. In the challenge, English children's audiobooks were provided as training data. For this challenge, we focused on: 1) automatic construction of a training corpus for TTS systems from audiobooks; 2) design of linguistic features for statistical parametric speech synthesis (SPSS) based on audiobooks; and 3) deep neural network-based SPSS. Large-scale subjective evaluation results show that the NITech system synthesized high natural and highest intelligible 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.

Typical TTS systems have two main components: text analysis and speech waveform generation. In the text analysis component, linguistic features, e.g., phonemes, syllables, accents, and parts-of-speech, of an input text is estimated. In the speech waveform generation component, a speech waveform is generated from the linguistic features estimated with the text analysis component. Corpus-based speech synthesis approaches, such as unit-selection [1] and statistical parametric speech synthesis (SPSS), have been proposed for the speech waveform generation component. SPSS, e.g., involving hidden Markov model- [2] and deep neural network (DNN)-based speech synthesis [3], has been actively investigated and the quality of synthetic speech has greatly improved.

Although many TTS systems have been proposed, comparisons of such systems are difficult when the task or corpus is 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 [4]. This challenge has so far provided English, Mandarin, some Indian languages, English audiobooks, etc. as training data. 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 expressive speech. 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 used 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 [5]. In the Blizzard Challenge 2016 (this year's challenge), about five hours of speech data from professionally produced English children's audiobooks were provided as training data [6]. All 50 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.

We focused on three approaches for this challenge: 1) automatic construction of a training corpus for TTS systems from audiobooks; 2) design of linguistic features for SPSS based on audiobooks; and 3) DNN-based SPSS. 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 an acoustic model of SPSS. To overcome this problem, we investigated the automatic construction of a training corpus from audiobooks using a speech recognizer. Children's audiobooks consist of descriptive and conversational parts. Speech data, especially in the conversational part, include various speaking styles, emotions, characters, etc. In SPSS, the definition of linguistic features is important to capture speech data diversity. Therefore, we designed linguistic features for children's audiobooks. Appropriate mapping from linguistic features to acoustic features is needed to synthesize high-quality speech. Recently, DNNs have been introduced to SPSS and have the potential to produce naturally sounding synthesized speech [3, 7, 8, 9, 10]. DNN-based acoustic models can represent complex mapping functions from linguistic features to acoustic features. For this challenge, we used a trajectory training method that takes into account the global variance in the DNN training [10].

The rest of this paper is organized as follows. Section 2 describes the Nagoya Institute of Technology TTS system for the Blizzard Challenge 2016. 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

Figure 1 gives an overview of the Nagoya Institute of Technology (NITech) text-to-speech (TTS) system for the Blizzard Challenge 2016. In the training part, linguistic and acoustic

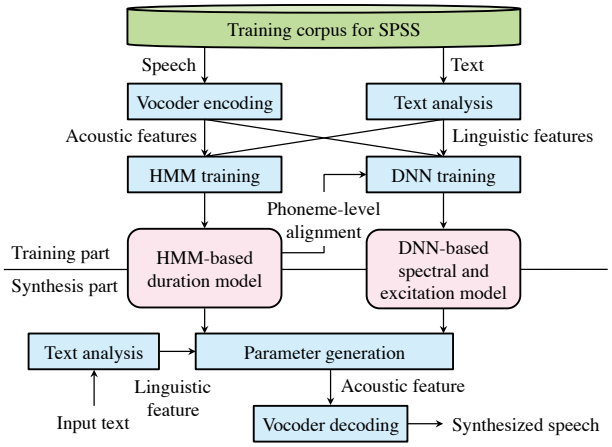

Figure 1: Overview of the NITech TTS system

features are first extracted from text analysis and vocoder encoding, respectively. Second, hidden Markov model (HMM)-based speech synthesizer is constructed to estimate phoneme-level alignments. Finally, deep neural network (DNN)-based speech synthesizer is constructed by using linguistic features, acoustic features, and phoneme-level alignments. In the synthesis part, acoustic features are estimated from linguistic features using the HMM-based duration and DNN-based spectral and excitation models. A synthesized speech is then generated from vocoder decoding. The details of the construction method of the training corpus, linguistic features for audiobooks, and DNN-based SPSS are described in the following sections.

2.1. Automatic construction of training corpus for TTS systems from audiobooks

Provided speech data of audiobooks include explicit page-turning sounds. These sounds are not suited for training acoustic models (AMs). To detect page-turning sounds, a Gaussian mixture model (GMM) is trained from some page-turning sounds. Speech data are divided into page-by-page speech data based on the detected page-turning sounds. After that, page-level, not sentence-level, training and synthesis are conducted in the NITech system.

Normally, some mismatches are present in speech data and text in audiobooks. It is preferable to use a text of fully matched speech data for the training corpus. We call the text of provided audiobooks *book text* and text of fully matched speech data *correct text*. It is expensive to manually obtain a large amount of correct texts. Therefore, a speech recognizer (SR) is used to estimate texts (*recognized text*) of speech data. A training corpus is composed of speech data and text pairs which achieved high word-match accuracy of book text and recognized text [11, 12]. We propose a method for estimating the correct text from a book and recognized text.

Figure 2 shows an overview of the training corpus construction method. It is assumed that most speech data match book texts. A language model (LM) based on book texts is very useful for speech recognition. Therefore, a book-adapted LM (BA-LM), which is adapted to each book, is used for speech recognition. A speaker-independent AM (SI-AM) and BA-LM are used for a speaker-independent SR (SI-SR) part. The adaptation corpus is composed of speech data and text pairs which achieved high word-match accuracy of book text and recognized text. In the speaker-adapted SR (SA-SR) part, a speaker-adapted AM (SA-AM) and BA-LM are used. Finally, a training corpus for SPSS is constructed from speech data and text pairs which achieved high word-match accuracy of book text and recognized text.

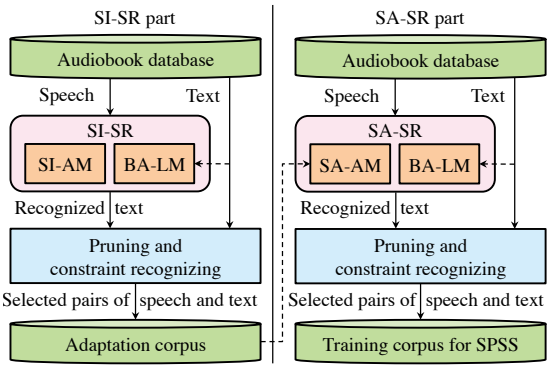

Figure 2: Overview of training corpus construction (SI: speaker independent, SA: speaker adapted, BA: book adapted, SR: speech recognizer, AM: acoustic model, LM: language model). Dashed arrows represent adaptation process.

Correct text: baby dinosaurs broke out of the eggs crack
Book text: baby dinosaurs broke out of the eggs
Recognized text: they dinosaurs broke out of the eggs crack

Figure 3: Examples of correct, book, and recognized texts

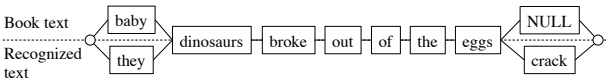

Figure 4: Example of word network

Some additional information, i.e., description of a book or onomatopoeia, which does not exist in book text, are recorded in the speech data of audiobooks. Figure 3 shows examples of correct, book, and recognized texts. In the examples of Figure 3, the onomatopoeia ("crack"), which does not exist in a book text, is recorded in the speech data. Since the text corresponding to the additional information does not exist, book texts negatively affect the training AMs. On the other hand, recognized texts may include speech recognition errors, e.g., "they" in Figure 3. To overcome mismatches of speech data and text, speech recognition using a constrained word network is conducted. Figure 4 shows an example of a word network consisting of book and recognized texts. The word network assigns a path penalty for the book text path if the book text is NULL; otherwise, it assigns a path penalty for the recognized text path. A SR with the word network without an LM is used to estimate texts (*constraint-recognized text*) of speech data. It is expected that constraint-recognized texts contain text corresponding to additional information and reducing speech recognition errors. Constraint-recognized texts are used as the texts of the training corpus for TTS systems.

2.2. Design of linguistic features for SPSS based on audiobooks

Speech parameters, such as spectrum, excitation, and duration, depend on a variety of contextual factors, e.g., phonemes, syllables, accents, and parts-of-speech. In SPSS, context-dependent

models are generally used to capture these contextual factors. If combinations of these contextual factors are taken into account, more accurate AMs can be obtained. Thus, appropriate context (linguistic features) design is needed to synthesize high-quality speech. Since the NITech TTS system performs page-level training and synthesis, we can use the linguistic features of phoneme, syllable, word, phrase, sentence, and page.

In audiobooks, the speech data in the conversational part of an audiobook are read emphatically, emotionally, and so on. On the other hand, the speech data in the descriptive part are read comparatively more neutrally than in the conversational part. Therefore, speech data in the conversational and descriptive parts should be distinguished by linguistic features. In the NITech TTS system, linguistic features based on double quotes are used to express the reading styles of speech data in the conversational and descriptive parts. In addition, natural speech includes prosodic information, such as intonation. To capture prosodic information, linguistic features of sentence-level parsing is used. The result of parsing is represented by a tree structure, which is called a syntactic tree. Some information obtained from the syntactic tree is used as linguistic features. We added linguistic features to the HTS English recipes [13] as follows:

- the number of {phrases, sentences} in this page
- position of the current sentence in this page
- whether the {previous, current, next} {phoneme, syllable, word, phrase} is enclosed by double quotes
- the rate of {word, phrase} enclosed by double quotes in this page
- guess part-of-speech of the parent of the current word
- distance on the syntactic tree between the current word and {the {previous, next} word, root of the syntactic tree, the {previous, next} content word}
- position of the current word in the parent of the current word
- the number of {phonemes, syllables, words} in the parent of the current word

2.3. DNN-based SPSS

In SPSS using DNN-based AMs [3, 7, 8], a single DNN is trained to represent a mapping function from linguistic features to acoustic features including spectral and excitation parameters with their dynamic features. In the generation process, 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 trajectory training considering global variance (GV) in the DNN training [10].

2.3.1. Standard DNN-based SPSS

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

$$\boldsymbol{o}_t = [\boldsymbol{c}_t^\top, \Delta^{(1)}\boldsymbol{c}_t^\top, \Delta^{(2)}\boldsymbol{c}_t^\top]^\top \tag{1}$$

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

$$\boldsymbol{o} = [\boldsymbol{o}_1^\top, \ldots, \boldsymbol{o}_t^\top, \ldots, \boldsymbol{o}_T^\top]^\top \tag{2}$$

$$\boldsymbol{c} = [\boldsymbol{c}_1^\top, \ldots, \boldsymbol{c}_t^\top, \ldots, \boldsymbol{c}_T^\top]^\top \tag{3}$$

where T is the number of frames included on a page. The relation between $\boldsymbol{o}$ and $\boldsymbol{c}$ can be represented as $\boldsymbol{o} = \boldsymbol{W}\boldsymbol{c}$, where $\boldsymbol{W}$ is a window matrix extending $\boldsymbol{c}$ to $\boldsymbol{o}$. The optimal static-feature vector sequence is obtained by

$$\hat{\boldsymbol{c}} = \arg\max_{\boldsymbol{c}} P(\boldsymbol{o}|\boldsymbol{\lambda}) = \arg\max_{\boldsymbol{c}} \mathcal{N}(\boldsymbol{W}\boldsymbol{c}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) \tag{4}$$

where $\boldsymbol{\lambda}$ is a parameter set and $\mathcal{N}(\cdot|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ denotes the Gaussian distribution with a mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. The $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are given by

$$\boldsymbol{\mu} = \left[\boldsymbol{\mu}_1^\top, \ldots, \boldsymbol{\mu}_t^\top, \ldots, \boldsymbol{\mu}_T^\top\right]^\top \tag{5}$$

$$\boldsymbol{\Sigma} = \mathrm{diag}\left[\boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\Sigma}_t, \ldots, \boldsymbol{\Sigma}_T\right] \tag{6}$$

The optimal static-feature sequence $\hat{\boldsymbol{c}}$ is given by

$$\hat{\boldsymbol{c}} = \boldsymbol{P}\boldsymbol{W}^\top\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}, \quad \boldsymbol{P} = \left(\boldsymbol{W}^\top\boldsymbol{\Sigma}^{-1}\boldsymbol{W}\right)^{-1} \tag{7}$$

As a result, smooth static-feature trajectories can be obtained using dynamic features as constraints. In DNN-based speech synthesis, the mean vector at frame t, $\boldsymbol{\mu}_t$, is obtained from a trained neural network and a linguistic-feature vector at time t, $\boldsymbol{l}_t$, as follows:

$$\boldsymbol{\mu}_t = g(\boldsymbol{l}_t|\boldsymbol{\lambda}_{\mathrm{NN}}) \tag{8}$$

where $g(\cdot|\boldsymbol{\lambda}_{\mathrm{NN}})$ is a non-linear mapping function represented by a neural network $\boldsymbol{\lambda}_{\mathrm{NN}}$. A covariance matrix is usually independent of linguistic features, i.e., a globally tied covariance matrix $\boldsymbol{\Sigma}_{\mathrm{G}}$ is used, in DNN-based speech synthesis.

Assuming that outputs of a neural network are used as mean parameters in a statistical model, an objective function can be defined as

$$\mathcal{L} = P(\boldsymbol{o}|\boldsymbol{\lambda}) = \mathcal{N}(\boldsymbol{o}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \prod_{t=1}^{T} \mathcal{N}(\boldsymbol{o}_t|\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_{\mathrm{G}}) \tag{9}$$

The parameter set $\boldsymbol{\lambda}$, which consists of the parameter of the neural network $\boldsymbol{\lambda}_{\mathrm{NN}}$ and the covariance matrix $\boldsymbol{\Sigma}_{\mathrm{G}}$, is optimized in the sense of maximum likelihood as follows:

$$\hat{\boldsymbol{\lambda}} = \arg\max_{\boldsymbol{\lambda}} P(\boldsymbol{o}|\boldsymbol{\lambda}) = \arg\max_{\boldsymbol{\lambda}} \prod_{t=1}^{T} \mathcal{N}(\boldsymbol{o}_t|\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_{\mathrm{G}}) \tag{10}$$

If an identity matrix is used as the covariance matrix, maximization of the objective function $\mathcal{L}$ is equivalent to minimization of the conventional frame-level mean square errors. The neural network can be trained by standard back-propagation using the gradient of the mean vector.

2.3.2. Trajectory training

In the standard DNN-based SPSS framework, although the frame-level objective function is used for DNN training, the sequence-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 DNNs.

The traditional likelihood function in Eq. (9) can be reformulated as a trajectory likelihood function by imposing the explicit relationship between static and dynamic features, which is given by $\boldsymbol{o} = \boldsymbol{W}\boldsymbol{c}$ [14]. The trajectory likelihood function of $\boldsymbol{c}$ is then written as

$$\mathcal{L}_{\mathrm{Trj}} = \frac{1}{Z} P(\boldsymbol{o}|\boldsymbol{\lambda}) = P(\boldsymbol{c}|\boldsymbol{\lambda}) = \mathcal{N}(\boldsymbol{c}|\bar{\boldsymbol{c}}, \boldsymbol{P}) \tag{11}$$

where Z is a normalization term. Inter-frame correlation is

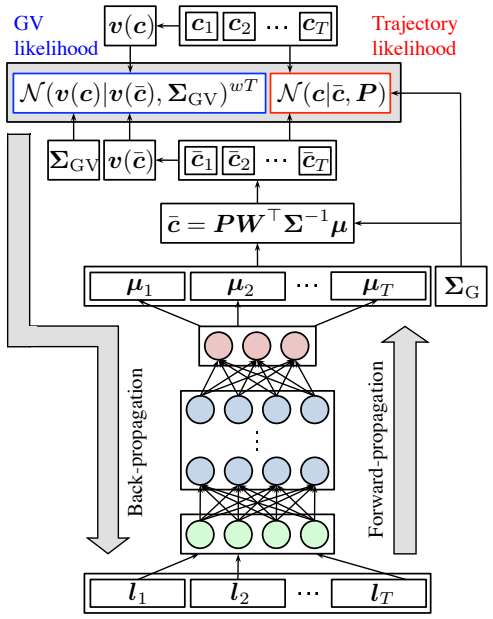

Figure 5: Overview of trajectory training considering GV for DNN-based SPSS

modeled by the covariance matrix $\boldsymbol{P}$ that is generally full. Note that the mean vector $\bar{\boldsymbol{c}}$ is equivalent to the generated static-feature sequence expressed by Eq. (7). The parameter set $\boldsymbol{\lambda}$ is estimated by maximizing the trajectory likelihood $\mathcal{L}_{\mathrm{Trj}}$.

2.3.3. *Trajectory training considering GV*

To address the over-smoothing problem of generated parameter trajectories, the concept of parameter generation considering the GV is introduced into the training of DNNs. Figure 5 shows an overview of trajectory training considering the GV. The objective function $\mathcal{L}_{\mathrm{GVTrj}}$ is given by

$$\begin{aligned}\mathcal{L}_{\mathrm{GVTrj}} &= P(\boldsymbol{c}|\boldsymbol{\lambda})P(\boldsymbol{v}(\boldsymbol{c})|\boldsymbol{\lambda},\boldsymbol{\lambda}_{\mathrm{GV}})^{wT} \\ &= \mathcal{N}(\boldsymbol{c}|\bar{\boldsymbol{c}},\boldsymbol{P})\mathcal{N}(\boldsymbol{v}(\boldsymbol{c})|\boldsymbol{v}(\bar{\boldsymbol{c}}),\boldsymbol{\Sigma}_{\mathrm{GV}})^{wT} \end{aligned} \quad (12)$$

where $\boldsymbol{v}(\boldsymbol{c}) = [v(1),\ldots,v(D)]^{\top}$ is a GV vector of the static-feature vector sequence $\boldsymbol{c}$. The GV vector is calculated page by page as follows:

$$v(d) = \frac{1}{T}\sum_{t=1}^{T}(c_t(d) - \langle c(d)\rangle)^2, \quad \langle c(d)\rangle = \frac{1}{T}\sum_{t=1}^{T} c_t(d) \quad (13)$$

where d is an index of the feature dimension. The mean vector of the probability density for the GV, $\boldsymbol{v}(\bar{\boldsymbol{c}})$, is defined as the GV of the mean vector of the trajectory likelihood function in Eq. (11), which is equivalent to the GV of the generated parameters expressed by Eq. (7). The GV likelihood $P(\boldsymbol{v}(\boldsymbol{c})|\boldsymbol{\lambda},\boldsymbol{\lambda}_{\mathrm{GV}})$ 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 $P(\boldsymbol{c}|\boldsymbol{\lambda})$ and $P(\boldsymbol{v}(\boldsymbol{c})|\boldsymbol{\lambda},\boldsymbol{\lambda}_{\mathrm{GV}})$ is controlled by the GV weight w.

The parameter set $\boldsymbol{\lambda}$, which consists of the parameter of the neural network $\boldsymbol{\lambda}_{\mathrm{NN}}$, the covariance matrix $\boldsymbol{\Sigma}_{\mathrm{G}}$, and the covariance matrix $\boldsymbol{\Sigma}_{\mathrm{GV}}$ of the GV vector, is estimated by maximizing the objective function $\mathcal{L}_{\mathrm{GVTrj}}$. The neural network can be updated and trained with by the back-propagation algorithm using the gradient of the mean vector $\boldsymbol{\mu}$. The parameters are optimized so that the GVs of generated trajectories get close to the natural ones.

The optimal static-feature vector sequence $\hat{\boldsymbol{c}}$ is determined by maximizing the objective function $\mathcal{L}_{\mathrm{GVTrj}}$ as follows:

$$\hat{\boldsymbol{c}} = \arg\max_{\boldsymbol{c}} P(\boldsymbol{c}|\boldsymbol{\lambda})P(\boldsymbol{v}(\boldsymbol{c})|\boldsymbol{\lambda},\boldsymbol{\lambda}_{\mathrm{GV}}) \quad (14)$$

Since this estimate is equivalent to the maximum likelihood estimate by using the basic parameter generation algorithm expressed by Eq. (4), the basic parameter generation algorithm can be used for this framework.

3. Blizzard Challenge 2016 evaluation

3.1. Training corpus construction conditions

The collection of provided children's audiobooks consisted of 50 books with a total 1090 pages. A SR was trained to construct a training corpus for SPSS. The CMU pronunciation dictionary [15] and the WSJ0, WSJ1 [16], and TIMIT [17] 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 [18] and SRILM [19] were used to construct the SR. The training recipe was the same as that of the HTK Wall Street Journal Training Recipe [20]. Thresholds of word-match accuracy for adaptation and training corpora were set to 90%. After pruning, the training corpus for SPSS consisted of 825 pages.

3.2. TTS system construction conditions

Linguistic features were extracted using Festival [21] and Stanford Parser [22]. 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. The acoustic-feature vectors were composed of 228 dimensions: 49-dimension STRAIGHT [23] mel-cepstral coefficients including the 0th coefficient, F_0, 24-dimension mel-cepstral analysis aperiodicity measures, and their first- and second-order derivatives. The RAPT [24], SWIPE' [25], and REAPER [26] tools were used for F_0 extraction. The HMM-based SPSS was constructed to estimate phoneme-level alignments. A five-state left-to-right context-dependent multi-stream multi-space probability distribution (MSD)-HSMM [27, 28, 29, 30, 31] without skip transitions was used as the AM. Each state output a probability distribution was composed of a spectrum, F_0, and aperiodicity streams. The spectrum and aperiodicity streams were modeled using single multi-variate 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 1-dimensional Gaussian distribution. The HTS [13] was used for constructing the HMM-based SPSS. In the DNN-based SPSS, the input feature was a 426-dimensional feature vector consisting of 423 linguistic features including binary features and numerical features for contexts and three duration features including the duration of the current phoneme and position of the current frame. The out-

Table 1: Evaluation results (paid participants)

System	Page domain							Sentence domain		SUS
	OI	PL	SP	ST	IN	EM	LE	NAT	SIM	WER
A	49 ± 7.9	48 ± 7.5	49 ± 7.8	49 ± 7.2	50 ± 7.0	48 ± 8.8	51 ± 6.5	4.9 ± 0.31	4.7 ± 0.65	–
B	26 ± 10.1	29 ± 11.0	24 ± 11.1	24 ± 11.4	26 ± 11.5	31 ± 11.2	25 ± 9.8	3.2 ± 1.05	3.5 ± 1.03	0.19
C	17 ± 7.7	16 ± 7.8	24 ± 10.8	21 ± 10.6	20 ± 10.3	21 ± 10.5	19 ± 8.9	1.8 ± 0.92	1.5 ± 0.87	0.18
D	25 ± 10.4	25 ± 10.7	30 ± 10.7	28 ± 10.7	26 ± 11.2	26 ± 12.2	27 ± 10.0	2.1 ± 1.01	1.9 ± 1.08	0.16
E	19 ± 8.7	22 ± 11.0	17 ± 9.5	19 ± 10.6	20 ± 11.1	26 ± 11.6	18 ± 8.1	2.3 ± 1.12	2.1 ± 1.03	0.32
$\boldsymbol{F}$	$\mathbf{24 \pm 9.8}$	$\mathbf{23 \pm 9.8}$	$\mathbf{30 \pm 12.1}$	$\mathbf{29 \pm 11.8}$	$\mathbf{27 \pm 12.0}$	$\mathbf{27 \pm 11.6}$	$\mathbf{27 \pm 9.5}$	$\mathbf{3.0 \pm 1.14}$	$\mathbf{2.6 \pm 1.11}$	**0.12**
G	19 ± 8.6	18 ± 8.2	23 ± 10.8	21 ± 10.2	20 ± 10.1	20 ± 10.0	20 ± 8.9	2.1 ± 0.96	1.8 ± 0.97	0.20
H	19 ± 9.5	24 ± 12.5	14 ± 8.3	18 ± 10.7	20 ± 11.4	26 ± 11.5	17 ± 8.4	2.6 ± 1.22	3.5 ± 1.18	0.43
I	16 ± 7.0	16 ± 7.6	22 ± 11.1	21 ± 10.3	19 ± 9.9	20 ± 10.2	18 ± 8.1	1.9 ± 0.92	1.6 ± 0.79	0.25
J	18 ± 8.0	20 ± 9.1	21 ± 10.9	21 ± 10.4	20 ± 10.6	25 ± 10.6	18 ± 8.1	2.1 ± 0.96	2.0 ± 1.07	0.27
K	18 ± 9.0	19 ± 9.6	25 ± 11.6	24 ± 10.8	22 ± 10.6	23 ± 11.4	20 ± 9.2	2.1 ± 1.01	1.4 ± 0.78	0.30
L	36 ± 11.0	38 ± 10.2	33 ± 11.4	33 ± 12.1	35 ± 11.2	36 ± 11.2	35 ± 10.9	4.1 ± 0.85	4.1 ± 0.85	0.15
M	33 ± 11.1	35 ± 11.4	31 ± 12.1	33 ± 12.1	34 ± 11.5	37 ± 11.2	33 ± 10.7	3.9 ± 0.96	3.9 ± 0.89	0.21
N	14 ± 7.2	15 ± 8.2	21 ± 11.1	18 ± 9.3	17 ± 9.6	20 ± 10.1	16 ± 7.7	1.8 ± 0.86	1.4 ± 0.73	0.45
O	19 ± 8.3	20 ± 9.0	27 ± 11.5	26 ± 10.8	23 ± 10.8	22 ± 10.7	22 ± 8.9	2.4 ± 1.03	2.1 ± 1.09	0.26
P	17 ± 8.0	20 ± 10.6	17 ± 11.0	18 ± 10.9	20 ± 11.1	25 ± 11.4	16 ± 8.4	2.1 ± 0.99	2.1 ± 1.09	0.43
Q	32 ± 11.3	33 ± 11.7	28 ± 11.7	27 ± 13.0	30 ± 12.6	34 ± 11.4	29 ± 10.4	3.7 ± 0.91	3.9 ± 0.97	0.25

put feature was a 229-dimensional feature vector consisting of a 228-dimension acoustic feature and voiced/unvoiced binary value. 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 unit-variance. The input and output features were time-aligned frame-by-frame by using the trained MSD-HSMM. A single network, which models both spectral and excitation parameters, was trained. The architecture of the DNNs was three hidden layers with 2048 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 DNNs, a mini-batch stochastic gradient descent (SGD)-based back-propagation algorithm was used. The GV weight w was set to 0.01.

3.3. Experimental conditions of listening test

Large-scale subjective listening tests were conducted by the Blizzard Challenge 2016 organization [32]. The listeners included paid participants, volunteers, and speech experts. The paid participants (104 participants and native speakers of English) took the test in soundproof listening booths using high-quality headphones. The volunteers and speech experts 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 scored 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 scored from 1 to 5 based on the following 2-sentence-domain-criteria.

- naturalness (NAT): "completely unnatural" to "completely natural"
- 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 1 lists the scores and standard deviations of the results from the paid participants. Systems A, B, C, D, and $\boldsymbol{F}$ represent the following systems.

- A: natural speech
- B: Festival benchmark system
- C: HTS benchmark system
- D: DNN benchmark system
- $\boldsymbol{F}$: NITech system

The page-domain results show that $\boldsymbol{F}$ ranked 6^{th}, 7^{th}, 3^{rd}, 3^{rd}, 4^{th}, 5^{th}, and 4^{th} out of the 16 TTS systems listed in Table 1 for page-domain-criteria OI, PL, SP, ST, IN, EM, and LE, respectively. The $\boldsymbol{F}$ system achieved high MOSs for SP, ST, IN, and LE. It is believed that page-level training and synthesis lead to a high MOS for SP. High MOSs for ST and IN were due to the linguistic features of parsing and trajectory training.

The sentence-domain results show that $\boldsymbol{F}$ ranked 5^{th} and 6^{th} for sentence-domain-criteria NAT and SIM, respectively. Each higher ranked system L, M, Q, and B obtained high MOSs for NAT and SIM. By contrast, compared with NAT and SIM with $\boldsymbol{F}$, there was a large difference between NAT (3.0) and SIM (2.6). Speaker similarity indicated the weakness of SPSS. Therefore, investigation of methods for improving a similarity is required. In terms of intelligibility, $\boldsymbol{F}$ achieved the lowest WER (0.12). It is believed that linguistic features of double quotes lead to stable synthesized speech.

4. Conclusion

We described the Nagoya Institute of Technology (NITech) text-to-speech (TTS) system for the Blizzard Challenge 2016. We investigated the automatic construction of a training corpus for TTS systems from audiobooks, design of linguistic features for statistical parametric speech synthesis (SPSS) based on audiobooks, and deep neural network (DNN)-based SPSS. Large-scale subjective evaluation results show that the NITech TTS system synthesized highly naturalness speech. However, it did not obtain good similarity. In terms of intelligibility, the system achieved the lowest WER. Future work will include improving similarity, investigating sentence-level control for DNN-based SPSS [33].

5. Acknowledgement

The research leading to these results was partly funded by the Core Research for Evolutional Science and Technology (CREST) from the Japan Science and Technology Agency (JST).

6. References

[1] A. Hunt and A. W. Black, "Unit selection in a concatenative speech synthesis system using a large speech database," *1996 IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 1, pp. 373–376, 1996.

[2] K. Tokuda, Y. Nankaku, T. Toda, H. Zen, J. Yamagishi, and K. Oura, "Speech synthesis based on hidden Markov models," *IEEE*, vol. 101, no. 5, pp. 1234–1252, 2013.

[3] H. Zen, A. Senior, and M. Schuster, "Statistical parametric speech synthesis using deep neural networks," *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 7962–7966, 2013.

[4] A. W. Black and K. Tokuda, "The Blizzard Challenge – 2005: Evaluating corpus-based speech synthesis on common datasets," *Interspeech 2005*, pp. 77–80, 2005.

[5] S. King and V. Karaiskos, "The blizzard challenge 2013," *Blizzard Challenge 2013 Workshop*, 2013.

[6] "Blizzard Challenge 2016," http://www.synsig.org/index.php/Blizzard_Challenge_2016.

[7] H. Lu, S. King, and O. Watts, "Combining a vector space representation of linguistic context with a deep neural network for text-to-speech synthesis," *ISCA SSW8*, pp. 281–285, 2013.

[8] Y. Qian, Y. Fan, H. Wenping, and F. Soong, "On the training aspects of deep neural network (DNN) for parametric TTS synthesis," *2014 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 3857–3861, 2014.

[9] K. Hashimoto, K. Oura, Y. Nankaku, and K. Tokuda, "The effect of neural networks in statistical parametric speech synthesis," *2015 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 4455–4459, 2015.

[10] ——, "Trajectory training considering global variance for speech synthesis based on neural networks," *2016 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 5600–5604, 2016.

[11] N. Braunschweiler, M. Gales, and S. Buchholz, "Lightly supervised recognition for automatic alignment of large coherent speech recordings," *Interspeech 2010*, pp. 2222–2225, 2010.

[12] S. Takaki, K. Sawada, K. Hashimoto, K. Oura, and K. Tokuda, "Overview of NITECH HMM-based speech synthesis system for Blizard Challenge 2013," *Blizzard Challenge 2013 Workshop*, 2013.

[13] "HTS," http://hts.sp.nitech.ac.jp/.

[14] H. Zen, K. Tokuda, and T. Kitamura, "Reformulating the HMM as a trajectory model by imposing explicit relationships between static and dynamic features," *Computer Speech and Language*, vol. 21, no. 1, pp. 153–173, 2007.

[15] "CMU pronouncing dictionary," http://www.speech.cs.cmu.edu/cgi-bin/cmudict.

[16] D. B. Paul and J. M. Baker, "The design for the wall street journal-based CSR corpus," *The workshop on Speech and Natural Language*, pp. 357–362, 1992.

[17] J. Garofolo, L. Lamel, W. Fisher, J. Fiscus, D. Pallett, N. Dahlgren, and V. Zue, "TIMIT: acoustic-phonetic continuous speech corpus," *Linguistic Data Consortium*, 1993.

[18] "HTK," http://htk.eng.cam.ac.uk/.

[19] "SRILM," http://www.speech.sri.com/projects/srilm.

[20] K. Vertanen, "Baseline WSJ acoustic models for HTK and Sphinx: Training recipes and recognition experiments," *Cavendish Laboratory*, 2006.

[21] "Festival," http://www.festvox.org/festival/.

[22] "Stanford Parser," http://nlp.stanford.edu/software/lex-parser.shtml.

[23] H. Kawahara, I. Masuda-Katsuse, and A. Cheveigne, "Restructuring speech representations using a pitch-adaptive time-frequency smoothing and an instantaneous-frequency-based F0 extraction: Possible role of a repetitive structure in sounds," *Speech Communication*, vol. 27, pp. 187–207, 1999.

[24] D. Talkin, "A robust algorithm for pitch tracking (RAPT)," *Speech Coding and Synthesis*, pp. 495–518, 1995.

[25] A. Camacho, "SWIPE: a sawtooth waveform inspired pitch estimator for speech and music," *Ph.D. Thesis, University of Florida*, 2007.

[26] "REAPER," https://github.com/google/REAPER.

[27] H. Zen, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Hidden semi-Markov model based speech synthesis," *8th International Conference on Spoken Language Processing*, pp. 1185–1180, 2004.

[28] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Simultaneous modeling of spectrum, pitch and duration in HMM-based speech synthesis," *Eurospeech 1999*, pp. 2347–2350, 1999.

[29] K. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi, and T. Kitamura, "Speech parameter generation algorithms for HMM-based speech synthesis," *2000 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 936–939, 2000.

[30] K. Tokuda, T. Masuko, N. Miyazaki, and T. Kobayashi, "Multi-space probability distribution HMM," *IEICE Transactions on Information & Systems*, vol. E85-D, no. 3, pp. 455–464, 2002.

[31] K. Shinoda and T. Watanabe, "MDL-based context-dependent subword modeling for speech recognition," *Acoustical Science and Technology*, vol. 21, no. 2, pp. 76–86, 2000.

[32] R. A. J. Clark, M. Podsiadlo, M. Fraser, C. Mayo, and S. King, "Statistical analysis of the blizzard challenge 2007 listening test results," *Blizzard Challenge 2007 Workshop*, 2007.

[33] O. Watts, Z. Wu, and S. King, "Sentence-level control vectors for deep neural network speech synthesis," *Interspeech 2015*, pp. 2217–2221, 2015.