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 English 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 Nagoya Institute of Technology (NITech) have 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.

We focused on three approaches for last year's challenge [6]: 1) automatic construction of a training corpus for SPSS systems from audiobooks; 2) design of linguistic features for SPSS based on audiobooks; and 3) deep neural network acoustic models incorporating trajectory training. For this year's challenge, we redesigned linguistic features for SPSS based on audiobooks. Features obtained from dependency parsing [7], types of sentences [8], and word and phrase codes were used as additional linguistic features from the last year's system. Moreover, we introduce the parameter trajectory generation process considering the global variance into the training [9] of mixture density network based acoustic models [10, 11].

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 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

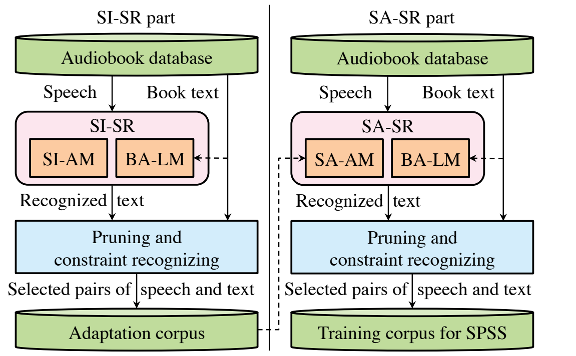

Figure 1: *Overview of training corpus construction (SI: speaker independent, SA: speaker adapted, BA: book adapted, SR: speech recognizer, AM: acoustic model, LM: language model)*

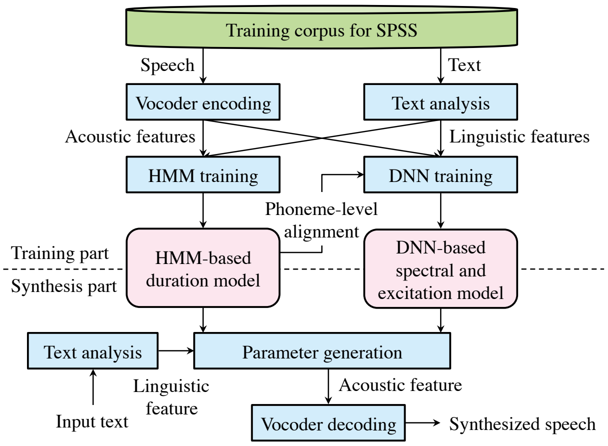

Figure 2: *Overview of the NITech TTS system*

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 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.

2.1. Design of linguistic features for SPSS based on audiobooks

Acoustic models are trained to estimate acoustic features from linguistic features. It is necessary to use appropriate linguistic features for the training corpus. However, conventional linguistic features are not designed assuming training corpus of audiobooks. Therefore, we redesign linguistic features for SPSS based on audiobooks. Table 1 lists the linguistic features for the NITech TTS system.

Information up to sentence-level has been used as conventional linguistic features. Since audiobooks are semantically related between sentences, linguistic features up to sentence-level are insufficient. We introduce page-level linguistic features because a scene changes by a page in children's audiobooks. Moreover, since sentence structures are related to prosody, linguistic features which capture sentence structures are useful for acoustic model training. However, conventional linguistic features are difficult to express complex sentence structures. 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 an angry phrase in the training corpus and the test phrase is high. If the phrase in the training corpus with the highest similarity is recorded with an angry style, the test phrase can be synthesized with the angry style. In this way, a speaking style can be selected automatically from the text of the test phrase. Like the phrase code, a word code is also introduced as linguistic features. The word2vec [14] is used to measure word similarity of a test word and training corpus words.

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 $\boldsymbol{o}_t$ 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 $\Delta^{(2)}\boldsymbol{c}_t$.

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

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

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

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

Table 1: *Linguistic features for SPSS. The bold font means additional linguistic features to the HTS-2.3.1 English demo script. The "parent" represents a node of syntactic tree and the "father" and "children" represent a node of dependency tree.*

Level	Details of linguistic features
Phoneme	The {phoneme before the previous / previous / current / next / phoneme after the next} phoneme identity. {Forward / backward} position of the current phoneme identity in the current syllable. **Whether the {previous / current / next} phoneme identity is enclosed by double quotes.**
Syllable	Whether the {previous / current / next} syllable {stressed / accented}. The number of phonemes in the {previous / current / next} syllable. The number of {stressed / accented} syllables {before / after} the current syllable in the current phrase. The number of syllables from the previous {stressed / accented} syllable to the current syllable. The number of syllables from the current syllable to the next {stressed / accented} syllable. {Forward / backward} position of the current syllable in the current {word / phrase}. Phoneme identity of the vowel of the current syllable. **Whether the {previous / current / next} syllable is enclosed by double quotes.**
Word	Guess part-of-speech of the {previous / current / next} word. The number of syllables in the {previous / current / next} word. The number of content words {before / after} the current word in the current phrase. The number of words from the {previous content word to the current word / current word to the next context word}. {Forward / backward} position of the current word in the current phrase. **Whether the {previous / current / next} word is enclosed by double quotes. Guess part-of-speech of the parent of the current word. The number of {phonemes / syllables / words} in the parent of the current word. {Forward / backward} position of the current word in the parent of the current word. Distance on the syntactic tree between the current word and the {previous word / next word / root of the syntactic tree / previous content word / next content word}. The current {word to the father / father word to the grandfather / grandfather word to the grandgrandfather } word relation. The number of children relations. Distance on the dependency tree between the current word and the {previous / next / root} word. Distance on the text between the current word to the {father / grandfather / grandgrandfather} word. Word code of the current word.**
Phrase	The number of {syllables / words} in the {previous / current / next} phrase. {Forward / backward} position of the current phrase in the current sentence. TOBI endtone of the current phrase. **Whether the {previous / current / next} phrase is enclosed by double quotes. Phrase code of the current phrase.**
Sentence	The number of {syllables / words / phrases} in the current sentence. **{Forward / backward} position of the current sentence in the current page. Type of the current sentence.**
Page	**The number of {phrases / sentences} in the current page. The rate of {words / phrases} enclosed by double quotes in the current page.**

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

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

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

$$\hat{\boldsymbol{c}} = \boldsymbol{P}\boldsymbol{W}^\top\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}, \quad \boldsymbol{P} = \left(\boldsymbol{W}^\top\boldsymbol{\Sigma}^{-1}\boldsymbol{W}\right)^{-1} \tag{5}$$

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

An MDN maps a linguistic-feature vector $\boldsymbol{l}$ 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

$$\mathcal{L} = P(\boldsymbol{o} \mid \boldsymbol{l}, \boldsymbol{\lambda}_{\mathrm{MDN}}) = \mathcal{N}(\boldsymbol{o} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \prod_{t=1}^{T} \mathcal{N}(\boldsymbol{o}_t \mid \boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t) \tag{6}$$

where the mean and covariance parameter are obtained by $\boldsymbol{\mu}_t = [\mu_{t,1}, \mu_{t,2}, \ldots, \mu_{t,D}]^\top$ and $\boldsymbol{\Sigma}_t = \mathrm{diag}[\sigma_{t,1}^2, \sigma_{t,2}^2, \ldots, \sigma_{t,D}^2]$, respectively. Then, the mean and standard deviation at frame t, $\mu_{t,d}$ and $\sigma_{t,d}$, can be obtained as follows:

$$\mu_{t,d} = g_d^{(\boldsymbol{\mu})}(\boldsymbol{l}_t, \boldsymbol{\lambda}_{\mathrm{MDN}}) \tag{7}$$

$$\sigma_{t,d} = \exp(g_d^{(\boldsymbol{\sigma})}(\boldsymbol{l}_t, \boldsymbol{\lambda}_{\mathrm{MDN}})) \tag{8}$$

where $g_d^{(\boldsymbol{\mu})}(\boldsymbol{l}_t, \boldsymbol{\lambda}_{\mathrm{MDN}})$ and $g_d^{(\boldsymbol{\sigma})}(\boldsymbol{l}_t, \boldsymbol{\lambda}_{\mathrm{MDN}})$ are the activations of the output layer corresponding to mean and standard deviation parameters, given $\boldsymbol{l}_t$ and $\boldsymbol{\lambda}_{\mathrm{MDN}}$, respectively. The MDN parameter set $\boldsymbol{\lambda}_{\mathrm{MDN}}$ is optimized in the sense of maximum likelihood as follows:

$$\begin{aligned}\hat{\boldsymbol{\lambda}}_{\mathrm{MDN}} &= \arg\max_{\boldsymbol{\lambda}_{\mathrm{MDN}}} P(\boldsymbol{o} \mid \boldsymbol{l}, \boldsymbol{\lambda}_{\mathrm{MDN}}) \\ &= \arg\max_{\boldsymbol{\lambda}_{\mathrm{MDN}}} \prod_{t=1}^{T} \mathcal{N}(\boldsymbol{o}_t \mid \boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)\end{aligned} \tag{9}$$

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 $\boldsymbol{o} = \boldsymbol{W}\boldsymbol{c}$ [16]. The trajectory likelihood function of $\boldsymbol{c}$ is then written as

$$\mathcal{L}_{\mathrm{Trj}} = \frac{1}{Z} P(\boldsymbol{o} \mid \boldsymbol{l}, \boldsymbol{\lambda}) = P(\boldsymbol{c} \mid \boldsymbol{l}, \boldsymbol{\lambda}) = \mathcal{N}(\boldsymbol{c} \mid \bar{\boldsymbol{c}}, \boldsymbol{P}) \tag{10}$$

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

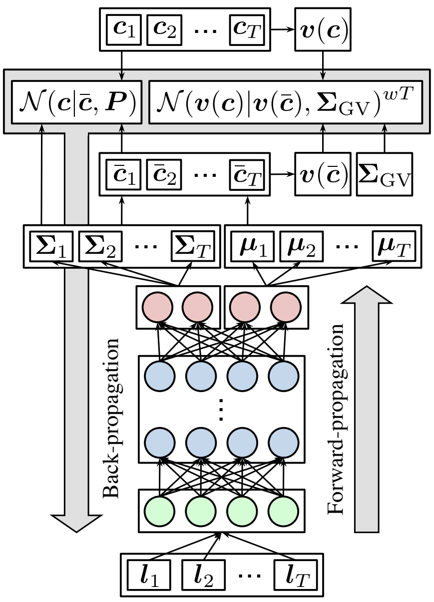

Figure 3: *Overview of trajectory training considering GV for MDN-based SPSS*

feature sequence expressed by Eq. (5). The parameter set $\boldsymbol{\lambda}$ is estimated by maximizing the trajectory likelihood $\mathcal{L}_{\mathrm{Trj}}$.

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 $\mathcal{L}_{\mathrm{GVTrj}}$ is given by

$$\begin{aligned}\mathcal{L}_{\mathrm{GVTrj}} &= P(\boldsymbol{c} \,|\, \boldsymbol{l}, \boldsymbol{\lambda}) P(\boldsymbol{v}(\boldsymbol{c}) \,|\, \boldsymbol{l}, \boldsymbol{\lambda}, \boldsymbol{\lambda}_{\mathrm{GV}})^{wT} \\ &= \mathcal{N}(\boldsymbol{c} \,|\, \bar{\boldsymbol{c}}, \boldsymbol{P}) \mathcal{N}(\boldsymbol{v}(\boldsymbol{c}) \,|\, \boldsymbol{v}(\bar{\boldsymbol{c}}), \boldsymbol{\Sigma}_{\mathrm{GV}})^{wT} \end{aligned} \quad (11)$$

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:

$$v(d) = \frac{1}{T}\sum_{t=1}^{T}(c_t(d) - \langle c(d) \rangle)^2, \;\; \langle c(d) \rangle = \frac{1}{T}\sum_{t=1}^{T} c_t(d) \quad (12)$$

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. (10), which is equivalent to the GV of the generated parameters expressed by Eq. (5). The GV likelihood $P(\boldsymbol{v}(\boldsymbol{c}) \,|\, \boldsymbol{l}, \boldsymbol{\lambda}, \boldsymbol{\lambda}_{\mathrm{GV}})$ 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 $P(\boldsymbol{c} \,|\, \boldsymbol{l}, \boldsymbol{\lambda})$ and $P(\boldsymbol{v}(\boldsymbol{c}) \,|\, \boldsymbol{l}, \boldsymbol{\lambda}, \boldsymbol{\lambda}_{\mathrm{GV}})$ is controlled by the GV weight w. The parameter set $\boldsymbol{\lambda}$, which consists of the parameter of the MDN $\boldsymbol{\lambda}_{\mathrm{MDN}}$ and the covariance matrix $\boldsymbol{\Sigma}_{\mathrm{GV}}$ of the GV vector, is estimated by maximizing the objective function $\mathcal{L}_{\mathrm{GVTrj}}$. 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:

$$\hat{\boldsymbol{c}} = \arg\max_{\boldsymbol{c}} P(\boldsymbol{c} \,|\, \boldsymbol{l}, \boldsymbol{\lambda}) P(\boldsymbol{v}(\boldsymbol{c}) \,|\, \boldsymbol{l}, \boldsymbol{\lambda}, \boldsymbol{\lambda}_{\mathrm{GV}}) \quad (13)$$

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 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% [6]. 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. Voting 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 [30] 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-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 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-

Table 2: *Evaluation results*

System	Page domain							Sentence domain		SUS
	OI	PL	SP	ST	IN	EM	LE	NAT	SIM	WER
A	$47 \pm 9^*$	$46 \pm 10^*$	$47 \pm 9^*$	$47 \pm 10^*$	$47 \pm 10^*$	$47 \pm 10^*$	$47 \pm 9^*$	$4.7 \pm 0.6^*$	$4.5 \pm 0.8^*$	–
I	$38 \pm 10^*$	$38 \pm 10^*$	$36 \pm 11^*$	$36 \pm 12^*$	$37 \pm 11^*$	$38 \pm 11^*$	$36 \pm 11^*$	$4.0 \pm 0.9^*$	$4.0 \pm 0.9^*$	34 ± 31
G	32 ± 10	31 ± 10	34 ± 11	33 ± 11	$33 \pm 10^*$	34 ± 11	32 ± 10	3.6 ± 0.9	3.1 ± 1.2	35 ± 34
$\boldsymbol{L}$	$\mathbf{31 \pm 10}$	$\mathbf{30 \pm 10}$	$\mathbf{31 \pm 12}$	$\mathbf{31 \pm 12}$	$\mathbf{31 \pm 11}$	$\mathbf{33 \pm 11}$	$\mathbf{31 \pm 10}$	$\mathbf{3.6 \pm 0.9}$	$\mathbf{3.0 \pm 1.1}$	$\mathbf{30 \pm 32}$
E	31 ± 12	32 ± 12	$27 \pm 13^*$	$27 \pm 13^*$	$28 \pm 13^*$	32 ± 12	$28 \pm 12^*$	3.5 ± 1.0	$3.9 \pm 0.9^*$	$38 \pm 32^*$
P	31 ± 11	31 ± 10	31 ± 11	32 ± 12	32 ± 11	34 ± 11	30 ± 10	3.4 ± 1.0	$3.6 \pm 1.0^*$	$38 \pm 32^*$
B	$27 \pm 11^*$	$27 \pm 11^*$	$25 \pm 12^*$	$26 \pm 12^*$	$27 \pm 12^*$	$30 \pm 11^*$	$25 \pm 11^*$	$3.3 \pm 1.0^*$	$3.7 \pm 1.0^*$	$42 \pm 31^*$
M	$26 \pm 10^*$	$25 \pm 10^*$	$24 \pm 11^*$	$27 \pm 11^*$	$26 \pm 11^*$	$26 \pm 11^*$	$25 \pm 10^*$	$3.2 \pm 0.9^*$	3.1 ± 1.0	33 ± 33
K	$26 \pm 10^*$	$27 \pm 10^*$	$26 \pm 12^*$	$25 \pm 12^*$	$26 \pm 11^*$	$28 \pm 11^*$	$25 \pm 10^*$	$3.1 \pm 1.0^*$	2.9 ± 1.1	$42 \pm 31^*$
Q	$25 \pm 10^*$	$26 \pm 10^*$	$23 \pm 11^*$	$23 \pm 12^*$	$24 \pm 11^*$	$28 \pm 11^*$	$23 \pm 10^*$	$3.1 \pm 1.1^*$	2.8 ± 1.1	$44 \pm 31^*$
D	$25 \pm 10^*$	$23 \pm 10^*$	30 ± 12	$28 \pm 12^*$	$26 \pm 11^*$	$24 \pm 11^*$	$26 \pm 10^*$	$2.7 \pm 0.9^*$	$2.5 \pm 1.0^*$	30 ± 34
H	$24 \pm 9^*$	$24 \pm 9^*$	$25 \pm 11^*$	$25 \pm 11^*$	$23 \pm 10^*$	$21 \pm 10^*$	$24 \pm 9^*$	$2.7 \pm 1.0^*$	$2.1 \pm 0.9^*$	$39 \pm 31^*$
J	$22 \pm 9^*$	$21 \pm 9^*$	$24 \pm 11^*$	$23 \pm 11^*$	$22 \pm 10^*$	$20 \pm 11^*$	$22 \pm 10^*$	$2.7 \pm 1.0^*$	$2.6 \pm 1.0^*$	35 ± 33
F	$21 \pm 10^*$	$22 \pm 10^*$	$24 \pm 11^*$	$24 \pm 11^*$	$24 \pm 11^*$	$25 \pm 11^*$	$21 \pm 10^*$	$2.6 \pm 1.0^*$	$2.4 \pm 1.0^*$	$46 \pm 29^*$
C	$16 \pm 8^*$	$15 \pm 8^*$	$21 \pm 11^*$	$19 \pm 10^*$	$18 \pm 10^*$	$18 \pm 10^*$	$17 \pm 9^*$	$1.7 \pm 0.7^*$	$1.9 \pm 0.9^*$	$38 \pm 32^*$
O	$11 \pm 7^*$	$11 \pm 7^*$	$22 \pm 12^*$	$19 \pm 12^*$	$16 \pm 11^*$	$16 \pm 10^*$	$11 \pm 7^*$	$1.5 \pm 0.7^*$	$1.5 \pm 0.7^*$	$60 \pm 28^*$
N	$8 \pm 6^*$	$8 \pm 6^*$	$16 \pm 11^*$	$13 \pm 11^*$	$12 \pm 9^*$	$11 \pm 8^*$	$7 \pm 6^*$	$1.1 \pm 0.5^*$	$1.2 \pm 0.5^*$	$85 \pm 19^*$

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 unit-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 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 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 2 lists the means and standard deviations of the listening test results from the all listeners. Systems A, B, C, D, and $\boldsymbol{L}$ represent the following systems.

- A: natural speech
- B: unit-selection benchmark system
- C: HMM benchmark system
- D: DNN benchmark system
- $\boldsymbol{L}$: NITech system

The ordering of systems is in descending order of NAT. Wilcoxon's signed rank tests were used to determine significance difference [37]. In Table 2, asterisk * means a statistically significant difference between system $\boldsymbol{L}$ and other systems.

The page-domain results show that system $\boldsymbol{L}$ ranked 4$^{\text{th}}$, 5$^{\text{th}}$, 3$^{\text{rd}}$, 4$^{\text{th}}$, 4$^{\text{th}}$, 4$^{\text{th}}$, and 3$^{\text{rd}}$ 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 I statistically significantly better than our system $\boldsymbol{L}$ except IN criterion. Overall, our system $\boldsymbol{L}$ achieved good performance. The sentence-domain results show that system $\boldsymbol{L}$ ranked 3$^{\text{rd}}$ and 7$^{\text{th}}$ for sentence-domain-criteria NAT and SIM, respectively. Our system $\boldsymbol{L}$ achieved naturally 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 $\boldsymbol{L}$ 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

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

[2] S. King, “Measuring a decade of progress in text-to-speech,” *Loquens*, vol. 1, no. 1, 2014.

[3] S. King and V. Karaiskos, “The blizzard challenge 2013,” *Blizzard Challenge 2013 Workshop*, 2013.

[4] ——, “The blizzard challenge 2016,” *Blizzard Challenge 2016 Workshop*, 2016.

[5] “Blizzard Challenge 2017,” http://www.synsig.org/index.php/Blizzard_Challenge_2017.

[6] K. Sawada, C. Asai, K. Hashimoto, K. Oura, and K. Tokuda, “The NITech text-to-speech system for the Blizzard Challenge 2016,” *Blizzard Challenge 2016 Workshop*, 2016.

[7] R. Dall, K. Hashimoto, K. Oura, Y. Nankaku, and K. Tokuda, “Redefining the linguistic context feature set for HMM and DNN TTS through position and parsing,” *Interspeech 2016*, pp. 2851–2855, 2016.

[8] L.-H. Chen, Y. Jiang, M. Zhou, Z.-H. Ling, and L.-R. Dai, “The USTC system for Blizzard Challenge 2016,” *Blizzard Challenge 2016 Workshop*, 2016.

[9] K. Hashimoto, K. Oura, Y. Nankaku, and K. Tokuda, “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.

[10] C. M. Bishop, “Mixture density networks,” *Aston University*, 1994.

[11] H. Zen and A. Senior, “Deep mixture density networks for acoustic modeling in statistical parametric speech synthesis,” *2014 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 3844–3848, 2014.

[12] 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.

[13] Q. Le and T. Mikolov, “Distributed representations of sentences and documents,” *31st International Conference on Machine Learning*, 2014.

[14] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” *arXiv:1301.3781*, 2013.

[15] 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.

[16] 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.

[17] “CMU Pronouncing Dictionary,” http://www.speech.cs.cmu.edu/cgi-bin/cmudict.

[18] 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.

[19] 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.

[20] “HTK,” http://htk.eng.cam.ac.uk/.

[21] “SRILM,” http://www.speech.sri.com/projects/srilm.

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

[23] “Festival,” http://www.festvox.org/festival/.

[24] “Stanford Parser,” http://nlp.stanford.edu/software/lex-parser.shtml.

[25] D. Andor, C. Alberti, D. Weiss, A. Severyn, A. Presta, K. Ganchev, S. Petrov, and M. Collins, “Globally normalized transition-based neural networks,” *arXiv:1603.06042*, 2016.

[26] R. Rehurek and P. Sojka, “Software framework for topic modelling with large corpora,” *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pp. 45–50, 2010.

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

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

[29] “REAPER,” https://github.com/google/REAPER.

[30] 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.

[31] 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.

[32] 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.

[33] 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.

[34] 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.

[35] “HTS,” http://hts.sp.nitech.ac.jp/.

[36] “SPTK,” http://sp-tk.sourceforge.net/.

[37] 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.

[38] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, “WaveNet: A generative model for raw audio,” *arXiv:1609.03499*, 2016.