PAPER

Constructing text-to-speech systems for languages with unknown pronunciations

Kei Sawada*, Kei Hashimoto, Keiichiro Oura,
Yoshihiko Nankaku and Keiichi Tokuda

*Department of Scientific and Engineering Simulation, Nagoya Institute of Technology,
Gokiso-cho, Showa-ku, Nagoya, 464–0856 Japan*

(*Received 10 April 2017, Accepted for publication 28 September 2017*)

Abstract: This paper proposes a method for constructing text-to-speech (TTS) systems for languages with unknown pronunciations. One goal of speech synthesis research is to establish a framework that can be used to construct TTS systems for any written language. Generally, language-specific knowledge is required to construct TTS systems for a new language. However, it is difficult to acquire language-specific knowledge in each new language. Therefore, constructing a TTS system for a new language entails huge costs. To address this problem, we investigate a framework for automatically constructing a TTS system from a target language database consisting of only speech data and corresponding Unicode texts. In the proposed method, pseudo phonetic information of the target language with unknown pronunciation is obtained by a speech recognizer of a rich-resource proxy language. Then, a grapheme-to-phoneme converter and a statistical parametric speech synthesizer are constructed based on the obtained pseudo phonetic information. The proposed method was applied to Japanese and was evaluated in terms of objective and subjective measures. Additionally, we challenged the construction of TTS systems for nine Indian languages using the proposed method, and TTS systems were evaluated in the Blizzard Challenge 2014 and 2015.

Keywords: Text-to-speech system, Statistical parametric speech synthesis, Unknown pronunciation language, Low-resource language, Language-independent method

PACS number: 43.72.Ja [doi:10.1250/ast.39.119]

1. INTRODUCTION

A number of studies on text-to-speech (TTS) systems have been conducted. Consequently, the quality of synthetic speech has improved, and TTS systems are now used in various applications, such as in-car navigation, spoken dialogue, and speech translation systems. Accordingly, the demand for TTS systems offering high-quality synthetic speech, various speaking styles, and various languages is increasing. There are thousands of active written languages in the world [1]. Construction of a TTS system for a new language leads to increased use of applications. TTS systems for low-resource languages are in great demand because speech translation systems are very useful applications for low-resource languages. However, conventional methods of constructing corpus-based TTS systems for a new language not only require preparation of training corpus but also require language-specific knowledge. Especially, to marshal language-specific knowledge about pronunciation for each new language requires high cost. Therefore, a goal of the speech synthesis research is to establish a language-independent framework that can be used to construct TTS systems for any written language.

TTS systems can be examined as a text-to-speech mapping problem. Phoneme, the simplest abstract class of speech sounds, is a widely used intermediate representation for mapping. Thus, TTS systems have two main components: text analysis (text-to-phoneme) and speech waveform generation (phoneme-to-speech). In the text analysis part, a phoneme of an input text is estimated by using a lexicon which contains phonetic information. Additionally, some phonetic contextual factors, e.g., accents and parts-of-speech, are also estimated. These phoneme and phonetic contextual factors are used linguistic features. Since this part is highly dependent on the target language, it is costly to construct a TTS system for someone not familiar with the target language. In the speech waveform generation part, a speech waveform is generated from the linguistic features estimated by the text analysis part. Corpus-based speech synthesis approaches such as unit-selection [2] and

*e-mail: swdkei@sp.nitech.ac.jp

statistical parametric speech synthesis (SPSS) have been proposed for the speech waveform generation part. SPSS, e.g., hidden Markov model (HMM)- and deep neural network (DNN)-based speech synthesis [3,4], has been actively researched and the quality of synthetic speech has greatly improved. An SPSS system has several advantages: 1) within its statistical training framework, it can train the statistical properties of speakers, speaking styles, emotions, etc. from a training corpus; 2) many techniques that were developed for HMM/DNN-based speech recognition can be applied to speech synthesis; and 3) multiple languages can easily be supported because the language-dependent element is the only set of linguistic features to be used.

To construct a TTS system for a new language, it is necessary to marshal language-dependent elements, e.g., to define a phoneset and linguistic features, such as accents and parts-of-speech, for each language. However, doing so requires language-specific knowledge. Therefore, a low language-dependency framework is needed in order to construct TTS systems for new languages. In this study, we focus on automatic construction of a TTS system without knowledge specific to the language with the unknown pronunciation. We construct a TTS system from a database consisting of the only speech data and Unicode [5] texts corresponding to speech data. The problem in this situation is that a phoneset, phonetic information corresponding to speech data, and a lexicon do not exist. To solve these phoneset and phonetic information problems, speech recognition is carried out by using the speech recognizer of a rich-resource proxy language. Pseudo phoneme sequences of the target language speech data are obtained from the speech recognition results. An SPSS-based speech synthesizer of the target language is then trained from speech data and pseudo phoneme sequence pairs. To solve the lexicon problem, we train a grapheme-to-phoneme converter based on joint-sequence models [6] from text and pseudo phoneme sequence pairs. The joint-sequence model is a N-gram model that models a joint-sequence in which grapheme and phoneme sequences are aligned. The model can estimate a phoneme sequence with the highest likelihood from a grapheme sequence. In addition, in order to improve quality of synthesized speech, we propose improvement of the speech recognizer and estimation of the phoneme sequence considering phoneme duration. With these processes, it becomes possible to construct a TTS system automatically without specific knowledge on the target language.

In another way to address language-dependency, several low language-dependency frameworks have been proposed [7–9]. Grapheme-based speech synthesis treat every single graphemes as separate phoneme [7,8]. Methods of constructing a TTS system based on UniTran [8], a transliteration framework to convert Unicode text into a guessed phoneme [10], and vector space models (VSMs) [9] have also been proposed. Unlike these low language-dependency methods, the proposed method can utilize the obtained pseudo phonetic information by the speech recognizer of a proxy language. Therefore, not only grapheme information but also phonetic information can be utilized to construct TTS systems in the proposed method.

We applied the proposed method to Japanese. The results of objective and subjective experiments are discussed and the impacts of components are analyzed. Comparing the proposed and grapheme-based TTS system, a subjective preference test was conducted. Additionally, we challenged the construction of TTS systems for nine Indian languages (Assamese, Bengali, Gujarati, Hindi, Malayalam, Marathi, Rajasthani, Tamil, and Telugu) using the proposed method in the Blizzard Challenge 2014 and 2015 [11,12]. The results of the Blizzard Challenge 2015 were shown that the proposed TTS system was more natural sounding than the baseline TTS system for many languages.

The rest of this paper is organized as follows. Section 2 describes the construction of TTS systems for languages with unknown pronunciations. The experimental conditions and results are given in Sects. 3 and 4. Section 5 presents the Blizzard Challenge 2015 evaluation results. Concluding remarks and an outline of future work are presented in the final section.

2. TEXT-TO-SPEECH SYSTEM CONSTRUCTION

2.1. Language-dependent Text-to-speech System Construction

Phonemes are widely used by text-to-speech (TTS) systems as intermediate representations for mapping a text to speech. The following language-dependent operations are needed in order to construct a TTS system of a new language.

- Define a phoneset and linguistic features.
- Construct a lexicon or grapheme-to-phoneme converter for the text analysis part.

Normally, the phoneset is defined based on the phonology of the target language. Linguistic features are designed based on pronunciation information obtained from texts of each language. Additionally, the lexicon for converting from graphemes to phonemes is manually created.

2.2. Constructing a Text-to-speech System for a Language with an Unknown Pronunciation

In this paper, we propose a method for constructing TTS systems that uses a target language database consisting of speech data and Unicode texts corresponding to speech data. In the case of an unknown-pronunciation language, it is difficult to define a phoneset and even more

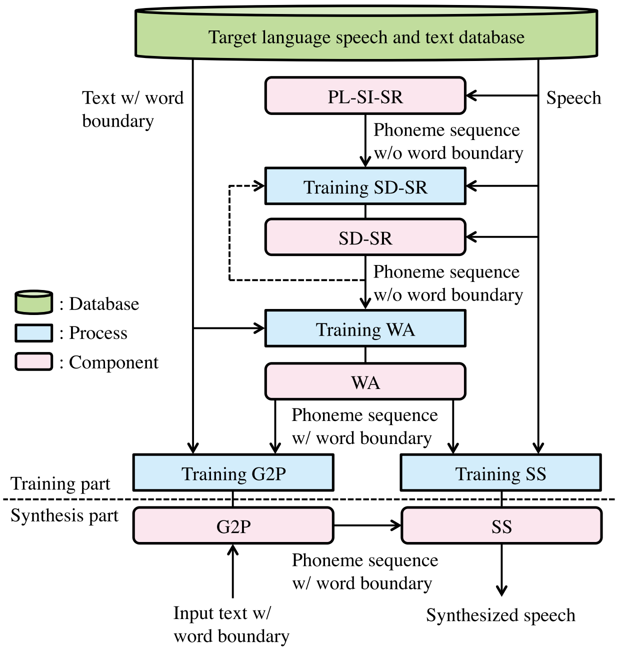

Target language speech and text database
Text w/ word boundary
PL-SI-SR
Speech
Phoneme sequence w/o word boundary
Training SD-SR
SD-SR
Phoneme sequence w/o word boundary
: Database
: Process
: Component
Training WA
WA
Phoneme sequence w/ word boundary
Training G2P
Training SS
Training part
Synthesis part
G2P
SS
Phoneme sequence w/ word boundary
Input text w/ word boundary
Synthesized speech

Fig. 1 Overview of the proposed TTS system construction method for a language with an unknown pronunciation (PL-SI-SR: proxy language speaker-independent speech recognizer, SD-SR: speaker-dependent speech recognizer, WA: word aligner, G2P: grapheme-to-phoneme converter, SS: speech synthesizer).

difficult to construct a hand-made lexicon, because they require manual operations used language-specific knowledge. Furthermore, it is hard to obtain a phoneme sequence corresponding to the speech data. To solve these problems, a speech recognizer of a rich-resource proxy language, e.g., English, for the target language can be used for automatic acquisition of phoneme sequences. The phoneset of the proxy language speech recognizer is then used as the phoneset of the target language. Although the phoneset is different from the appropriate phoneset of the target language, similar phonemes are assigned to speech data in this approach. To overcome the lexicon problem, a grapheme-to-phoneme converter based on a statistical model is used instead of a hand-made lexicon. In this way, entire TTS systems can be constructed within a statistical framework.

Figure 1 shows an overview of the proposed TTS system construction method for a language with an unknown pronunciation. This method consists of a speech recognizer (SR), word aligner (WA), grapheme-to-phoneme converter (G2P), and speech synthesizer (SS). The details of each component are described in the following sections.

2.2.1. Speech recognizer (SR)

In the case of SPSS, phoneme sequences corresponding to the speech data are necessary for acoustic modeling. To obtain phoneme sequences, speech recognition is carried out by using a proxy language speaker-independent SR (PL-SI-SR). For the target language recognition, a lexicon and language model are not used, and a phoneme network is designed so that each phoneme connects to every phoneme. In this way, the PL-SI-SR can work without being affected by a proxy-language-dependent phoneme sequence.

Since the accuracy of the phoneme sequences affects the latter components, i.e., the WA, G2P, and SS, it is important to estimate phoneme sequences accurately. To do so, a speaker-dependent SR (SD-SR) is constructed from initial phoneme sequences obtained by the PL-SI-SR. Furthermore, the phoneme sequence estimation and SD-SR training are iterated in order to adapt an acoustic model to training data. These iterations are acoustic-driven unsupervised training of speech units that uses the phoneset of the proxy language as an initial value.

Modeling of phoneme durations is important component for the SS. It is expected that phoneme sequences that are suitable for the SS can be obtained by taking account of phoneme duration. However, a hidden Markov model (HMM)-based SR has trouble accounting for phoneme duration because an HMM does not have explicit state duration information. Therefore, phoneme sequences are rescored using an alignment likelihood of a hidden semi-Markov model (HSMM) that has explicit state duration probability distributions. The phoneme sequence with the highest HSMM alignment likelihood in the N-best hypotheses of the HMM speech recognition result is selected as the pseudo phoneme sequence corresponding to the speech data.

2.2.2. Word aligner (WA)

Since many languages, e.g., English and Spanish, are written with spaces between words, a word-level G2P is suitable for the text analysis part. Furthermore, word boundary information is useful as linguistic features of the SS. However, a phoneme sequence obtained by the SR does not include word boundaries. Therefore, we construct a WA based on a joint-sequence model [6] for estimating word boundaries.

The optimal grapheme and phoneme pair alignment $\hat{\boldsymbol{w}}$ is estimated as follows:

$$\hat{\boldsymbol{w}} = \arg\max_{\boldsymbol{w}\in\boldsymbol{W}} P(\boldsymbol{w}). \qquad (1)$$

Here, $\boldsymbol{w}$ is a alignment of grapheme and phoneme pairs and $\boldsymbol{W}$ denotes the set of alignments of all possibly different grapheme and phoneme pairs. The parameters of the joint-sequence models are estimated by using the expectation-maximization (EM) algorithm. Pairs of texts with word boundaries and phoneme sequences obtained by the speech recognition are used for training. The WA is trained by providing a constraint condition such that a pause in the recognition results must be a word boundary. The Viterbi algorithm is used to align the grapheme and phoneme pairs.

The word boundary of the phoneme sequence are estimated by the phoneme corresponding to the grapheme with the word boundary.

2.2.3. Grapheme-to-phoneme converter (G2P)

To synthesize an arbitrary text, an input text needs to be converted into a phoneme sequence. However, in a language with an unknown pronunciation, it is difficult to construct a hand-made lexicon for converting input texts into phonemes. To overcome this problem, a G2P based on a joint-sequence model [6] is used instead of a hand-made lexicon. The G2P is trained from word-level pairs of text and phoneme sequences obtained by the SR and WA.

Insertion of appropriate pauses is important for natural synthesized speech. To estimate pauses by the G2P, word-level phoneme sequences of training data contain pauses in the speech recognition results. This makes it possible to estimate pauses when converting a phoneme sequence by the G2P.

2.2.4. Speech synthesizer (SS)

In the case of SPSS, context-dependent models are used to capture a variety of phonetic contextual factors. To generate naturally sounding synthesized speech, appropriate phonetic contextual factors (linguistic features) need to be defined. Here, we can use linguistic features of phoneme, syllable, word, phrase, and utterance. The details of these hierarchical linguistic features are as follows.

- Phoneme:
  - the current phoneme;
  - preceding and succeeding two phonemes;
  - the position of the current phoneme within the current syllable.
- Syllable:
  - the number of phonemes within preceding, current, and succeeding syllables;
  - the position of the current syllable within the current word and phrase;
  - the vowel identity within the current syllable.
- Word:
  - the number of syllables within preceding, current, and succeeding words;
  - the position of the current word within the current phrase.
- Phrase:
  - the number of syllables and words within preceding, current, and succeeding phrases;
  - the position of the current phrase within the utterance.
- Utterance:
  - the number of syllables, words, and phrases in the utterance.

A linguistic feature related to phoneme is obtained by the results of the SR. A syllable which is normally defined as C^*VC^* is useful as linguistic features of the SS. Here, C is a consonant, V is a vowel, and C^* indicates there may be none or more consonants. The consonant or vowel of a phoneme is dependent on the phoneset of the language used in the PL-SI-SR. A linguistic feature related to word is obtained by the results of the WA. A pause in the speech recognition results is defined as a phrase boundary. The SS can be constructed using the same procedure as the standard one from speech data and linguistic features corresponding to speech data.

3. EXPERIMENTAL CONDITIONS

3.1. Target Language Database Conditions

Objective and subjective experiments were conducted to evaluate the effectiveness of the proposed method. Since we can easily gather Japanese native subjects, listening tests for Japanese synthesized speech are desirable. Thus, Japanese was chosen as the target language. Of the 503 phonetically balanced sentences in the ATR Japanese speech database B-set [13] that were uttered by a male speaker MHT, 450 sentences were used for training and the remaining 53 sentences were used for testing.

Since there are a large number of graphemes in Japanese, e.g., hiragana, katakana, romaji, and kanji, a large amount of training data is needed to construct a G2P. Katakana, romaji and kanji can be represented by using hiragana in Japanese. Only hiragana was used as the graphemes in the experiments. Furthermore, assuming languages written with spaces between words, e.g., English and Spanish, a bunsetsu boundary which is a boundary of basic grammatical unit in Japanese was assumed as a word boundary in linguistic features. Table 1 shows an example of Japanese text for the experiments.

3.2. Speech Recognizer Conditions

An English SI-SR was used as the PL-SI-SR. The CMU pronunciation dictionary [14] and the WSJ0, WSJ1 [15], and TIMIT [16] databases were used to train the English SI-SR. The phoneset of English SI-SR has 40 phonemes. 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 comprised of 12-dimensional mel-frequency cepstral coefficients (MFCCs) including the 0th energy coefficient with the first- and second-order derivatives. A triphone three-state left-to-right Gaussian mixture model (GMM)-HMM without skip transitions was used as an acoustic model. The trained GMMs had 32 mixtures for pause and 16 mixtures for the other phonemes. The HTK [17] was used to construct the SR. The training procedures and model structures were the same as that of the HTK Wall Street Journal Training Recipe [18].

To consider phoneme duration, a five-state left-to-right monophone multi-stream multi-space probability distribu-

Table 1 Example of Japanese text in the experiment.

Original Japanese text	テレビゲームやパソコンでゲームをして遊ぶ
Japanese text for the experiment	てれびげーむや　ぱそこんで　げーむを　して　あそぶ

tion (MSD)-HSMMs [19–22] without skip transitions was trained from the TIMIT database. The other model structure and acoustic feature vector were the same as the SS.

3.3. Word Aligner and Grapheme-to-phoneme Convert Conditions

A joint-sequence model based WA was constructed from texts with word boundary and phoneme sequences without word boundary. The WA considered the context independent joint uni-gram.

A G2P based on the joint-sequence model was constructed from word-level pairs of text and phoneme sequence obtained by the SR and WA. As a result of a preliminary experiment, a joint eight-gram was used for the G2P structure. The G2P was trained by using the Sequitur G2P [23].

3.4. Speech Synthesizer Conditions

The speech signals were sampled at 16 kHz and windowed with a fundamental frequency (f_o)-adaptive Gaussian window with a 5-ms shift. The acoustic feature vectors were comprised of 183 dimensions: 39-dimension STRAIGHT [24] mel-cepstral coefficients including the 0th coefficient, $\log f_o$, 19-dimension mel-cepstral analysis aperiodicity measures including the 0th coefficient, and their first- and second-order derivatives. A five-state left-to-right context-dependent multi-stream MSD-HSMMs [19–22,25] without skip transitions was used as the acoustic model. Each state output distribution was composed of a spectrum, f_o, and aperiodicity streams. The spectrum and aperiodicity streams were modeled using single multi-variate Gaussian distributions with diagonal covariance matrices. The f_o 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. A parameter generation algorithm considering the global variance (GV) was applied [26]. The HTS [27] was used for constructing the SS.

A syllable is normally defined as C^*VC^* in the phonology. However, it is difficult to construct a language-independent method for estimating syllables. Therefore, in this experiment, a syllable is defined as C^*V assuming a Japanese mora which is basically one hiragana grapheme.

Table 2 MCD for synthesized speech obtained various insertion penalty in **1-best** and **50-best**.

Insertion penalty	MCD [dB]	
	1-best	**50-best**
−25	6.27	6.26
−20	6.28	6.22
−15	6.33	6.20
−10	6.27	6.27
−5	6.36	6.22
0	6.32	6.37
Average	6.31	6.26

4. EXPERIMENTAL RESULTS

4.1. Effect of Speech Recognizer

First, the effect of SR was experimentally evaluated. In the proposed method, speech recognition results affect components in the latter part. Therefore, the phoneme sequences obtained from the SR have a big impact on the quality of the synthetic speech.

To objectively evaluate the effect of rescoring using HSMM alignment likelihood, mel-cepstral distortions (MCDs) were calculated [28]. The PL-SI-SR (English SI-SR) was used to estimate pseudo phoneme sequences. Table 2 shows the results of MCDs in open data. **1-best** system did not apply HSMM-based rescoring process, i.e., speech recognition results with HMMs were used as the phoneme sequences of the training data. On the other hand, **50-best** system rescored the 50-best hypotheses, which were obtained from HMM-based speech recognition system, using the HSMM alignment likelihoods. From Table 2, since **50-best** system achieved lower average MCD than **1-best** system, the effectiveness of rescoring using HSMM alignment likelihood was confirmed. Consequently, **50-best** system was used to estimate phoneme sequences in the following experiments.

The effects of the phoneme insertion penalty and the number of iterations of training and recognition were investigated. A speech recognition score is calculated by using an acoustic model likelihood and a phoneme insertion penalty. The acoustic model likelihood tends to increase with a sequence of multiple short phonemes. The phoneme insertion penalty is a penalty parameter to control the number of phonemes included in speech recognition results. Figure 2 shows the average number of phonemes per sentence in each system. In Fig. 2, **PL-SI-SR** means

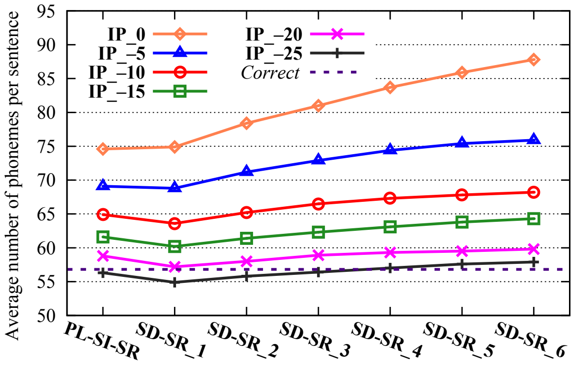


Fig. 2 Average number of phonemes per sentence in the training data. *Correct* means correct average number of phonemes using Japanese phoneset.

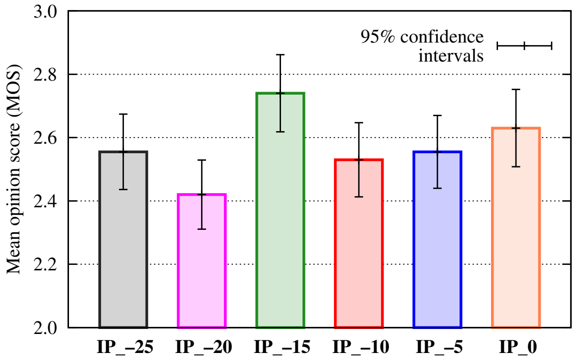
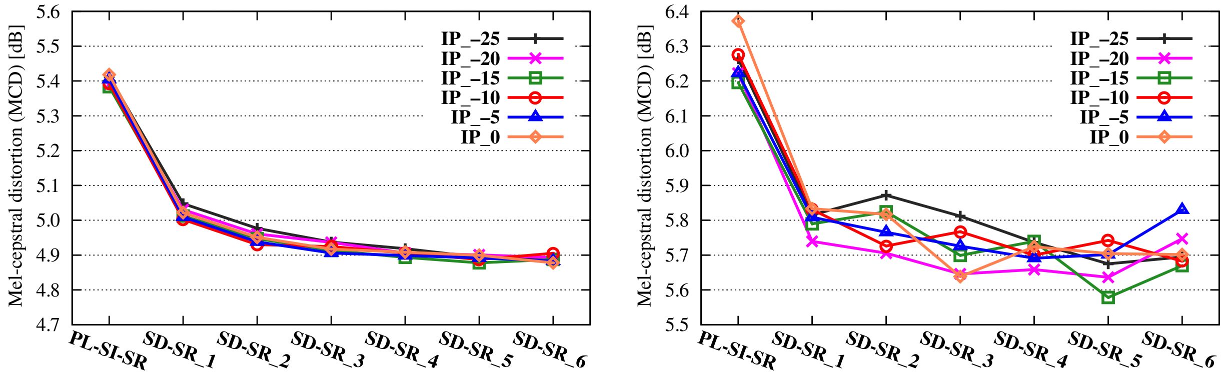


Fig. 4 MOS of naturalness with 95% confidence intervals for various insertion penalties in **SD-SR_5**.



(a) closed

(b) open

Fig. 3 MCD for synthesized speech obtained various iteration counts of the training and recognition.

systems using phoneme sequences obtained by the English SI-SR, **SD-SR_*i*** means systems using phoneme sequences obtained by the SD-SR (***i*** denotes iteration count), and **IP_*p*** means the phoneme insertion penalty (***p*** denotes value of phoneme insertion penalty). It is confirmed that the number of phonemes was influenced by the phoneme insertion penalty. The average number of phonemes increased with each iteration ***i***. In the proposed method, since the acoustic model was adapted to training data by iterations of training and recognition, the acoustic model likelihood increased with each iteration. Therefore, the influence of the phoneme insertion penalty relatively decreased and the average number of phonemes increased.

To objectively evaluate the effect of the phoneme insertion penalty and the number of iterations of training and recognition, MCDs were calculated. Figure 3 shows the results of MCDs in closed and open data. It can actually be seen in Fig. 3 that **SD-SR_*i*** systems achieved significantly lower MCDs than the **PL-SI-SR** system. For the **SD-SR_*i*** systems, MCD decreased as the number of iterations ***i*** increased. Despite the convergence of the MCDs in the closed data, the MCDs of the **SD-SR_6** systems became higher than those of the **SD-SR_5** systems in the open data. This is because the **SD-SR_6** systems had an overfitting problem.

A five-point mean opinion score (MOS) listening test with **SD-SR_5** having various insertion penalties was conducted in order to subjectively evaluate the naturalness of the synthesized speech. The subjects were ten Japanese students in our research group. All experiments were carried out using headphones in a soundproof room. For comparison, 20 sentences were chosen at random from the 53 test sentences. Speech samples were presented in random order for each test sentence. The scale of naturalness ran from 5 for "completely natural" to 1 for "completely unnatural" in the MOS test. The results of the MOS listening test are depicted in Fig. 4. It can be seen from the figure that **IP_–15** performed best. From Fig. 2, the number of phonemes in **IP_–15** was larger than the correct number of phonemes using the Japanese phoneset. These results suggest that the proposed system compensated for the differences in the phoneset by acoustic-driven short speech units. However, **IP_–10**, **IP_–5**, and **IP_0** did not obtain a higher MOS than the system **IP_–15**, though

Table 3 An example of phoneme sequences with word boundaries in training data (ぶんしょは　ねんねん　ふえていく). Where, | represent a word boundary.

System	Phoneme sequence with word boundaries
IP_–25	n b l iy s ih l ay | n eh n n ey n | f r ih p dh iy sp k uw
IP_–20	n b w iy zh ih l aa | n eh n n ey ng f r | ih p dh iy t k uw
IP_–15	n b uh ey s jh ih l aa | n eh n n ey n | f r ih p dh iy t k uw
IP_–10	n b uh ey s jh ih l ay | n eh n n ey ng d f r | ih p dh ey iy t k uw p
IP_–5	n b w oy iy s jh ih l aa | n eh n n ey ng | f r iy eh p dh iy iy t p g uw t
IP_0	n b uh ey ng z s jh ih l aa | n ey eh n n ey ng v f r | ey eh p dh iy iy t p g uw ih p

Table 4 Subjective preference scores.

Grapheme	**E-SI-SR**	Neutral	*p*-value
41.5%	57.0%	1.5%	0.0325

these systems included the large number of phonemes. Therefore, appropriate setting of phoneme insertion penalty is required to obtain high natural speech. Table 3 shows an example of phoneme sequences with word boundaries in training data obtained by **SD-SR_5** systems. It is confirmed that the pseudo phoneme sequence of the system with little influence of phoneme insertion penalty, such as **IP_0** and **IP_–5**, was composed of acoustic-driven short speech units. In addition, it can be seen that **IP_–20**, **IP_–10**, and **IP_0** contained errors in second word boundary in the example.

4.2. Comparing the Proposed and Grapheme-Based Systems

Grapheme-based speech synthesis system is often used as a baseline system for language-independent methods. Comparing the proposed and grapheme-based systems, a subjective preference listening test was conducted. The condition of preference listening test was the same as the MOS test. Table 4 shows the preference test result. **Grapheme** means a system which uses graphemes as speech unit instead of phonemes, and **E-SI-SR** means a system using the **50-best**, **SD-SR_5**, and **IP_–15** in Sect. 4.1. It can be seen from Table 4 that **E-SI-SR** was preferred to **Grapheme**. For this reason, the proposed method (**E-SI-SR**) may be useful for constructing a TTS system of a language with an unknown pronunciation without using language-specific knowledge. Since mappings from grapheme to phoneme are mostly unique in Japanese hiragana, **Grapheme** was able to synthesize speech with small pronunciation errors. In a language in which it is difficult to map from grapheme to phoneme, the proposed method is more expected to improve the performance compared to **Grapheme**.

4.3. Impact of Components

The proposed method estimates all linguistic features in the training and synthesis parts. To analyze the impact of each component, systems using correct linguistic features were compared. Additionally, a Japanese SI-SR was constructed by using the JNAS database [29] for comparison with the English SI-SR. The phoneset of Japanese SI-SR has 35 phonemes. The acoustic feature vector and model structure were the same as the English ones. Table 5 summarizes the compared systems and the following is a description of the compared systems.

- **Oracle**: system using correct linguistic features in the training and synthesis parts.
- **PhonemeWB**: system using correct linguistic features in the training part.
- **Phoneme**: system using correct phoneme sequences in the training part.
- **J-SI-SR**: system using a Japanese SI-SR, **50-best**, **SD-SR_5**, and **IP_–25**. The phoneme insertion penalty was set approximately to the correct number of phonemes.

To objectively evaluate the impact of components, MCD and root mean squared error (RMSE) of $\log f_0$ were used. Table 6 lists the results of the objective evaluation. In terms of MCD, the systems closer to **Oracle** obtained a lower MCD. There was a large difference in MCD between **Oracle** and **PhonemeWB**. Phoneme error rate (PER) of the G2P in **PhonemeWB** can be calculated because it uses correct phoneme sequences and word boundaries in the training part. To evaluate pause insertion accuracy, PER excluding pauses was also calculated. The G2P in **PhonemeWB** obtained a PER of 3.40% and a PER excluding pauses of 0.31%. Most of phoneme estimation errors of the G2P in **PhonemeWB** were caused by pause insertion errors. This result suggests that pause insertion errors have strong impacts on MCD and improvement of the G2P, especially pause insertion, is necessary to improve MCD. Error rates of the WA can be also calculated in **Phoneme**. The number of error boundaries included in the training data was one and the word boundary error rate was 0.04%. Therefore, the impact of the WA was not large comparing with the G2P in this experiment. From MCDs in Table 6, it can be seen that there was also a difference between **Phoneme** and **J-SI-SR**. This result indicates that

Table 5 Systems using correct linguistic features.

System	Phoneset	Training part		Synthesis part	Language of PL-SI-SR	Construction component
		Phoneme seq.	Word boundary	Phoneme seq.		
Oracle	Japanese	Correct	Correct	Correct	—	SS
PhonemeWB	Japanese	Correct	Correct	Estimate	—	G2P, SS
Phoneme	Japanese	Correct	Estimate	Estimate	—	WA, G2P, SS
J-SI-SR	Japanese	Estimate	Estimate	Estimate	Japanese	SR, WA, G2P, SS
E-SI-SR	English	Estimate	Estimate	Estimate	English	SR, WA, G2P, SS

Table 6 MCD and RMSE for synthesized speech of systems using correct linguistic features.

System	MCD [dB]	RMSE [log Hz]
Oracle	5.01	0.140
PhonemeWB	5.35	0.189
Phoneme	5.35	0.193
J-SI-SR	5.49	0.196
E-SI-SR	5.58	0.198

Table 7 GER of systems using correct linguistic features.

System	GER [%]
Oracle	5.73
PhonemeWB	6.69
Phoneme	5.54
J-SI-SR	22.52
E-SI-SR	33.33

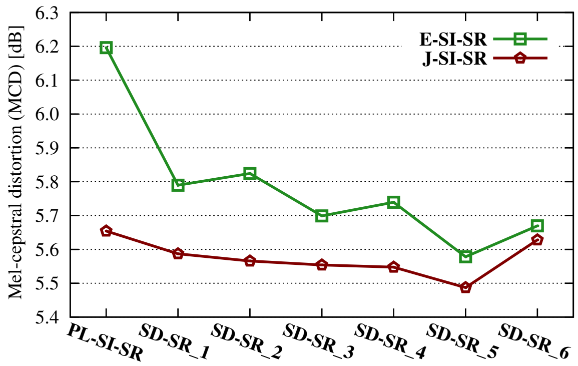


Fig. 5 MCD for synthesized speech obtained **E-SI-SR** and **J-SI-SR**.

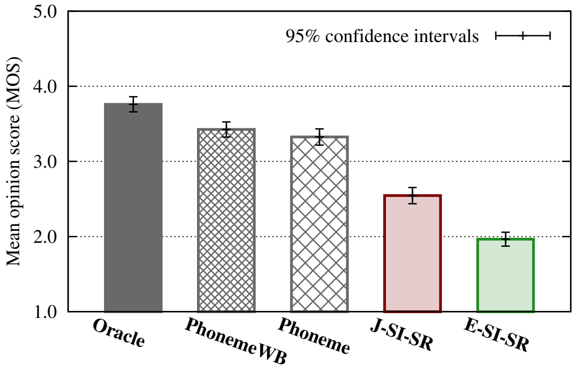


Fig. 6 MOS of naturalness with 95% confidence intervals for systems using correct linguistic features.

there is a difference between the correct phoneme sequence and pseudo phoneme sequence. Accordingly, improving speech recognition accuracy is necessary. Comparing **J-SI-SR** with **E-SI-SR**, there was a relatively large gap of MCDs. Figure 5 shows MCD for synthesized speech obtained **E-SI-SR** and **J-SI-SR**. Although speaker adaptation was applied from **PL-SI-SR** to **SD-SR_1** in **J-SI-SR**, the improvement of MCD was small. On the other hand, in **E-SI-SR**, speaker and language adaptation was applied from **PL-SI-SR** to **SD-SR_1**, and the MCD was significantly improved. This indicates that language adaptation is more effective than speaker adaptation in the proposed method. Additionally, from Table 6, RMSE showed the similar tendency as MCD.

To subjectively evaluate the impact of components, a five-point MOS listening test was conducted. Figure 6 shows the MOS of naturalness. As in the case of the objective evaluation in Table 6, the systems closer to **Oracle** obtained a higher MOS. There was a large difference in MOS between **Phoneme** and **J-SI-SR** and between **J-SI-SR** and **E-SI-SR**. These results indicate that speech recognition accuracy and phoneset of speech recognizer affect naturalness of synthetic speech in the proposed method.

Moreover, to evaluate intelligibility, intelligibility test was conducted. The subjects were asked to transcribe semantically unpredictable sentences (SUSs) by typing in the sentence they heard. 100 SUSs with each four words from the JEITA standard [30] were used for the evaluation. The subjects were ten Japanese students in our research group. Each subject typed 100 SUSs of a system chosen randomly. The average grapheme error rate (GER) was calculated from these transcripts. Table 7 lists the results of the intelligibility test in terms of GER. **Oracle**, **PhonemeWB**, and **Phoneme**, which used the phoneset based on the phonology, achieved low GER. Like the MOS evaluation in Fig. 6, there was a large difference in GER

Table 8 Number of native paid listeners.

Language	Number of native paid listeners
Bengali	48
Hindi	69
Malayalam	72
Marathi	69
Tamil	70
Telugu	70

Table 9 MOS of naturalness and speaker similarity in the Blizzard Challenge 2015.

Language	MOS of naturalness		MOS of similarity	
	Base	**NITech**	**Base**	**NITech**
Bengali	2.2	2.5	2.5	3.1
Hindi	3.2	2.3	2.6	2.8
Malayalam	1.6	1.7	1.8	2.3
Marathi	2.7	2.2	2.3	2.5
Tamil	2.2	2.4	1.8	2.3
Telugu	1.9	2.1	2.1	3.1

between **Phoneme** and **J-SI-SR** and between **J-SI-SR** and **E-SI-SR**. Ambiguous pronunciations had a bad influence on the GER. In **E-SI-SR**, several words were partially missing phonemes due to estimation errors in the G2P. The cause of these errors was the G2P training with training data including word boundary errors, such as word boundary errors in Table 3. In the case of **Phoneme** which used correct phoneme sequences, GER (5.54%) and word boundary error rate (0.04%) were low. Therefore, it is necessary to develop a noise-robust WA and improve the SR. It is considered that the low intelligibility influenced the low naturalness of **J-SI-SR** and **E-SI-SR**. In the future, we should investigate methods to improve intelligibility.

5. BLIZZARD CHALLENGE 2015 EVALUATION

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 [31]. The task of the Blizzard Challenge 2015 is constructing TTS systems for six Indian languages (Bengali, Hindi, Malayalam, Marathi, Tamil, and Telugu) [32]. These Indian languages have millions of speakers. However, these languages do not have a lot of resources for constructing a TTS system. The challenge is to construct TTS systems in each Indian language from the provided speech data sampled at 16 kHz and the corresponding Unicode text. About four or two hours of speech data in each of the six Indian languages are provided. To evaluate the synthesized speech, large-scale subjective evaluation tests were conducted by organizers of the Blizzard Challenge 2015. Table 8 summarizes the number of native paid listeners. We participated in the Blizzard Challenge 2015 [12] using the proposed method in this paper.

Table 9 shows results of five-point MOS tests in the read text task of the Blizzard Challenge 2015. In Table 9, **Base** means a baseline system used language-specific knowledge which was constructed by organizers using the FestVox [33] in the unit selection framework and **NITech** means our system. **NITech** systems were constructed without using language-specific knowledge based on Sect. 2.2. and system conditions were the same as Sects. 3.2., 3.3., and 3.4. Iteration count i and phoneme insertion penalty were adjusted for each language. Since Hindi has relatively rich-resource for constructing a TTS system in Indian languages, **Base** achieved higher MOS of naturalness than **NITech**. By contrast, **NITech** obtained higher MOSs of naturalness than **Base** in Bengali, Malayalam, Tamil, and Telugu. Furthermore, **NITech** achieved higher MOSs of speaker similarity than **Base** in all languages. For this reason, the proposed method is useful for constructing a TTS system of low-resource languages.

6. CONCLUSIONS

This paper has presented automatic construction of a text-to-speech (TTS) system from a target language database consisting of only speech data and corresponding Unicode texts. A grapheme-to-phoneme converter and speech synthesizer were constructed from speech recognition results of a proxy language speech recognizer. We applied this method to Japanese and evaluated the naturalness of its output. Experimental results showed that an appropriate phoneme insertion penalty and iteration count for training and recognition were important for the proposed method. The proposed TTS system that does not use language-specific knowledge could synthesize more natural speech compared with that from a grapheme-based TTS system. To improve the proposed method, the impact of each component was analyzed. The results suggest that pause insertion accuracy, speech recognition accuracy, and phoneset of speech recognizer affected objective measures.

Additionally, we applied the proposed method to six Indian languages. Subjective experiments of the Blizzard Challenge 2015 showed that the proposed system achieved higher naturalness than a baseline system of unit selection framework in four languages out of six languages. In terms of speaker similarity, the proposed system outperformed the baseline system in all languages.

Future work will include a multilingual speaker-independent speech recognizer based on the international phonetic alphabet (IPA) [34] or GlobalPhone [35] to obtain accurate phoneme sequences. Furthermore, investigations of prosodic attributes, e.g. accent, stress, and tone, and languages not written with space between words, e.g. Mandarin, Japanese, and Thai, will be needed in order to establish a more language-independent method. Additionally, we will perform experiments on various written languages.

ACKNOWLEDGMENTS

This research was supported by the Core Research for Evolutional Science and Technology (CREST) from the Japan Science and Technology Agency (JST).

REFERENCES

[1] Ethnologue, https://www.ethnologue.com/enterprise-faq/how-many-languages-world-are-unwritten-0 (accessed 2018-01-19).

[2] A. Hunt and A. W. Black, "Unit selection in a concatenative speech synthesis system using a large speech database," *Proc. ICASSP 1996*, pp. 373–376 (1996).

[3] K. Tokuda, Y. Nankaku, T. Toda, H. Zen, J. Yamagishi and K. Oura, "Speech synthesis based on hidden Markov models," *Proc. IEEE*, **101**, 1234–1252 (2013).

[4] H. Zen, A. Senior and M. Schuster, "Statistical parametric speech synthesis using deep neural networks," *Proc. ICASSP 2013*, pp. 7962–7966 (2013).

[5] Unicode Consortium, *The Unicode Standard: World-Wide Character Encoding* (Addison-Wesley, Boston, 1991).

[6] M. Bisani and H. Ney, "Joint-sequence models for grapheme-to-phoneme conversion," *Speech Commun.*, **50**, 434–451 (2008).

[7] O. Watts, J. Yamagishi and S. King, "Letter-based speech synthesis," *Proc. SSW7*, pp. 317–322 (2010).

[8] S. Sitaram, A. Parlikar, G. K. Anumanchipalli and A. W. Black, "Universal grapheme-based speech synthesis," *Proc. Interspeech 2015*, pp. 3360–3364 (2015).

[9] O. Watts, "Unsupervised learning for text-to-speech synthesis," *Ph.D. Thesis*, *University of Edinburgh* (2012).

[10] T. Qian, K. Hollingshead, S. Yoon, K. Kim and R. Sproat, "A python toolkit for universal transliteration," *Proc. LREC 2010*, pp. 2897–2901 (2010).

[11] K. Sawada, S. Takaki, K. Hashimoto, K. Oura and K. Tokuda, "Overview of NITECH HMM-based text-to-speech system for Blizzard Challenge 2014," *Proc. Blizzard Challenge 2014 Workshop* (2014).

[12] K. Sawada, K. Hashimoto, K. Oura and K. Tokuda, "The NITECH HMM-based text-to-speech system for the Blizzard Challenge 2015," *Proc. Blizzard Challenge 2015 Workshop* (2015).

[13] A. Kurematsu, K. Takeda, Y. Sagisaka, S. Katagiri, H. Kuwabara and K. Shikano, "ATR Japanese speech database as a tool of speech recognition and synthesis," *Speech Commun.*, **9**, 357–363 (1990).

[14] CMU Pronouncing Dictionary, http://www.speech.cs.cmu.edu/cgi-bin/cmudict (accessed 2018-01-19).

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

[16] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, N. L. Dahlgren and V. Zue, "TIMIT: Acoustic-phonetic continuous speech corpus," *Linguistic Data Consortium* (1993).

[17] HTK, http://htk.eng.cam.ac.uk/ (accessed 2018-01-19).

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

[19] H. Zen, K. Tokuda, T. Masuko, T. Kobayashi and T. Kitamura, "Hidden semi-Markov model based speech synthesis," *Proc. ICSLP 2004*, pp. 1185–1180 (2004).

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

[21] K. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi and T. Kitamura, "Speech parameter generation algorithms for HMM-based speech synthesis," *Proc. ICASSP 2000*, pp. 936–939 (2000).

[22] K. Tokuda, T. Masuko, N. Miyazaki and T. Kobayashi, "Multi-space probability distribution HMM," *IEICE Trans. Inf. Syst.*, **E85-D**, 455–464 (2002).

[23] Sequitur G2P, http://www-i6.informatik.rwth-aachen.de/web/Software/g2p.html (accessed 2018-01-19).

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

[25] T. Shinoda and T. Watanabe, "MDL-based context-dependent subword modeling for speech recognition," *Acoust. Sci. & Tech.*, **21**, 76–86 (2000).

[26] T. Toda and K. Tokuda, "Speech parameter generation algorithm considering global variance for HMM-based speech synthesis," *Proc. Interspeech 2005*, pp. 2801–2804 (2005).

[27] HTS, http://hts.sp.nitech.ac.jp/ (accessed 2018-01-19).

[28] T. Toda, A. W. Black and K. Tokuda, "Voice conversion based on maximum-likelihood estimation of spectral parameter trajectory," *IEEE Trans. Audio Speech Lang. Process.*, **15**, 2222–2235 (2007).

[29] K. Ito, M. Yamamoto, K. Takeda, T. Takezawa, T. Matsuoka, T. Kobayashi, K. Shikano and S. Itahashi, "JNAS: Japanese speech corpus for large vocabulary continuous speech recognition research," *J. Acoust. Soc. Jpn.* (*E*), **20**, 199–206 (1999).

[30] Technical Standardization Committee on Speech Input/Output Systems, "Speech synthesis system performance evaluation methods," Japan Electronic Industry Development Association (2003) (in Japanese).

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

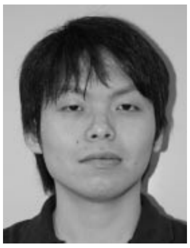
[32] K. Prahallad, A. Vadapalli, S. K. Rallabandi, S. Kesiraju, H. Murthy, T. Nagarajan, B. Singh, T. Sajani, K. S. Rao, S. V. Gangashetty, S. King, K. Tokuda and A. W. Black, "The Blizzard Challenge 2015," *Proc. Blizzard Challenge 2015 Workshop* (2015).

[33] FestVox, http://festvox.org/bsv/x3528.html (accessed 2018-01-19).

[34] International Phonetic Association, *Handbook of the International Phonetic Association: A Guide to the Use of the International Phonetic Alphabet* (Cambridge University Press, Cambridge, 1999).

[35] T. Schultz, "GlobalPhone: A multilingual speech and text database developed at Karlsruhe University," *ICSLP 2002*, pp. 345–348 (2002).

Kei Sawada received the B.E. and M.E. degrees in Computer Science and Scientific and Engineering Simulation from Nagoya Institute of Technology, Nagoya, Japan, in 2011 and 2013. He is currently a Ph.D. candidate at Nagoya Institute of Technology. His research interests include image recognition, speech recognition and synthesis, and machine learning.



Kei Hashimoto received the B.E., M.E., and Ph.D. degrees in computer science, computer science and engineering, and scientific and engineering simulation from Nagoya Institute of Technology, Nagoya, Japan in 2006, 2008, and 2011, respectively. From April 2012, he is now an Assistant Professor at Nagoya Institute of Technology, Nagoya, Japan. His research interests include statistical speech synthesis and speech recognition.

Keiichiro Oura received his Ph.D. degree in Computer Science and Engineering from Nagoya Institute of Technology, Nagoya, Japan, in 2010. He is currently a postdoctoral fellow of the CREST project at the Nagoya Institute of Technology. He received the ISCSLP Best Student Paper Award, the IPSJ YAMASHITA SIG Research Award, the ASJ ITAKURA Award, and IPSJ KIYASU Special Industrial Achievement Award, in 2008, 2010, 2013, and 2013, respectively. His research interests include statistical speech recognition and synthesis.

Yoshihiko Nankaku received his B.E. degree in Computer Science, and his M.E. and Ph.D. degrees in the Department of Electrical and Electronic Engineering from Nagoya Institute of Technology, Nagoya, Japan, in 1999, 2001, and 2004 respectively. After a year as a postdoctoral fellow at the Nagoya Institute of Technology, he became an Associate Professor at the same Institute. His research interests include statistical machine learning, speech recognition, speech synthesis, image recognition, and multi-modal interface.

Keiichi Tokuda received the B.E. degree in electrical and electronic engineering from Nagoya Institute of Technology, Nagoya, Japan, the M.E. and Dr.Eng. degrees in information processing from the Tokyo Institute of Technology, Tokyo, Japan, in 1984, 1986, and 1989, respectively. From 1996 to 2004 he was a Associate Professor at the Department of Computer Science, Nagoya Institute of Technology as Associate Professor, and now he is a Professor at the same institute. He is a IEEE Fellow and ISCA Fellow. His research interests include speech coding, speech synthesis and recognition, and statistical machine learning.