RECORDING SCRIPT DESIGN FOR CORPUS-BASED TTS SYSTEM BASED ON COVERAGE OF VARIOUS PHONETIC ELEMENTS

Mitsuaki ISOGAI, Hideyuki MIZUNO, Kazunori MANO

NTT Cyber Space Laboratories, NTT Corporation
1-1 Hikari-no-oka, Yokosuka-Shi, Kanagawa, 239-0847, Japan

ABSTRACT

This paper describes a new recording script generation method that can create speech databases for corpus-based TTS systems. This method is efficient due to its two features; (1) It has a 2-stage algorithm to generate the recording script with consideration of the balance of triphone, syllable and morpheme elements. (2) It can control types of phonetic elements included in the recording script via the weight coefficients of the phonetic elements. An evaluation shows that the 2-stage algorithm is effective in raising the coverage of phonetic elements and that this method yields a recording script containing various phonetic elements. A preference test shows that changing the selection criteria influences the quality of the synthesized speech. The same test also shows that it is better to take account of morpheme-based elements than syllable-based elements in generating a task-specific recording script.

1. INTRODUCTION

Recently a lot of text-to-speech(TTS) systems based on the corpus-based concatenative approach have been developed and shown to synthesize natural sounding speech[1-3]. The speech database is a key component of all corpus-based TTS systems. To construct a speech database, we must record a narrator's natural speech. To synthesize high-quality speech, the database must contain a wide variety of speech parts: words, syllables, and phonemes. If the script used in recording the parts is random or unbalanced, the recorded data may be full of redundancies and lack critical phonetic elements. Moreover, the script should be as small as possible because recording and labeling costs are very high.

Several methods to generate scripts have already reported [4-8]. For example, one exchanges sentence pairs using the entropy of diphones and triphones[4], another maximizes the synthesis unit coverage by taking account of prosody[6], and another uses multi-stage selection based on a greedy algorithm[9] using hit-rate and covering-rate for sentence selection criteria[7]. These methods use acoustic speech parts such as diphones, triphones, and syllable-based units.

In this paper, we propose a new script generation method that mines a large text corpus to automatically generate comprehensive and low redundancy scripts that are as small as possible. For high quality speech synthesis, the recording script should offer comprehensive word variety. Therefore, we take account of the balance of acoustic speech parts and linguistic ones. The acoustic parts provide variations in short-time speech features, while the linguistic parts provide long-time speech features such as words. Our method uses five phonetic elements: triphone, two acoustic parts and two linguistic parts. Our 2-stage selection algorithm raises the coverage of several key types of phonetic elements.

Section 2 defines the five phonetic elements and describes text corpus analysis. Section 3 introduces our script generation method. Section 4 details an experiment on script generation. Section 5 provides results of a listening test. Our conclusion is given in Section 6.

2. DEFINITIONS AND TEXT CORPUS ANALYSIS

2.1 Phonetic element definitions

In this paper, we define five phonetic elements as shown in Table 1. Triphone(represented by TRI) is a basic element to ensure that any text can be synthesized. The four expansion elements provide high-quality synthesis. Of the four expansion

Table 1. Phonetic element definitions. In the form column, C means consonant, V means vowel, P means any phoneme, S means syllable, and M means morpheme. '( )' means the enclosed phoneme is a phoneme environment. In the example column, '|' means a morpheme boundary. /AMAGUMO/ and /GA/ are Japanese morphemes.

	phonetic element	symbol	form	example
basic element	triphone	TRI	(C)V(C) (V)C(V) etc.	(/M/)/A/(/S/) (/A/)/K/(/A/)
expansion elements	syllable with phoneme environment	S1	(P)S(P)	(/O/)/NO/(/K/)
	syllable bigram with phoneme environment	S2	(P)SS(P)	(/O/)/NOKO/(/T/)
	morpheme with phoneme environment	M1	(P)M(P)	(/A/)/AMAGU-MO/(/G/)
	morpheme bigram with phoneme environment	M2	(P)MM(P)	(/A/)/AMAGU-MO/|/GA/(/A/)

elements, two are acoustic elements and the other two are linguistic ones. The acoustic elements are 'Syllable with phoneme environment'(S1) and 'syllable bigram with phoneme environment'(S2). The linguistic elements are 'Morpheme with phoneme environment'(M1) and 'morpheme bigram with phoneme environment'(M2).

2.2 Measurement definition

We define the metric of coverage to select sentences from a large text corpus. Let us define $Ec(X)$ as 'Element Cover Rate(ECR)' based on the total number of variations of the phonetic elements in the text corpus. Let us define $Tc(X)$ as 'Text Cover Rate(TCR)' based on frequencies of the phonetic elements in the text corpus. The definitions of $Ec(X)$ and $Tc(X)$ are as follows:

$$Ec(X) = \frac{M(X)}{N(X)} \tag{1}$$

$$Tc(X) = \frac{\sum_{i=1}^{N(X)} n_i(X)\, d_i(R)}{\sum_{i=1}^{N(X)} n_i(X)} \tag{2}$$

where X is the phonetic element type such as TRI. $N(X)$ is the number of variations of X in a text corpus C. $M(X)$ is the number of variations of X in a recording script R. $n_i(X)$ is the frequency of $u_i(X)$ in C. $\{u_1(X),\ldots,u_i(X),\ldots,u_{N(X)}(X)\}$ are phonetic elements included in C. Function $di(R)$ outputs 1 if $u_i(X) \in R$, otherwise it outputs 0.

2.3 Text corpus analysis

A recording script is generated by extracting sentences from a large text corpus. Our objective is to generate a script that has high coverage, as well as small size. If two scripts have the same size, the one with the higher coverage is preferred.

In 2.1, we defined TRI as a basic element. It is desirable that the script contains all TRIs in the text corpus. Some phonetic elements defined in 2.1 include other types of phonetic elements. For example, M2 can include TRI, S1, S2, and M1. This means that if we generate a script using only one phonetic element type in the selection criterion, other types of phonetic elements are also contained in the selected script without any further consideration.

We examined what percentage of the TRIs could be collected when an expansion element is used for text selection. The script generation algorithm in this analysis(see below) is based on a greedy algorithm[9].

Step1. The scores of all sentences in the text corpus are calculated. This score is defined as the increase in $Ec(X)$ or $Tc(X)$ that would occur if the sentence were added to the script.

Step2. The sentence with the highest score is selected from the text corpus. This sentence is added to the script and removed from the text corpus. Iterate from step1 to step2 until all types of TRIs in the text corpus are contained in the script.

Table 2. Contents of the text corpus.

genre		newspaper, newscast, novel
number of sentences		153479
number of mora		7168733
number of element variety in the text corpus	TRI	6537
	S1	65799
	S2	649054
	M1	579747
	M2	1593010

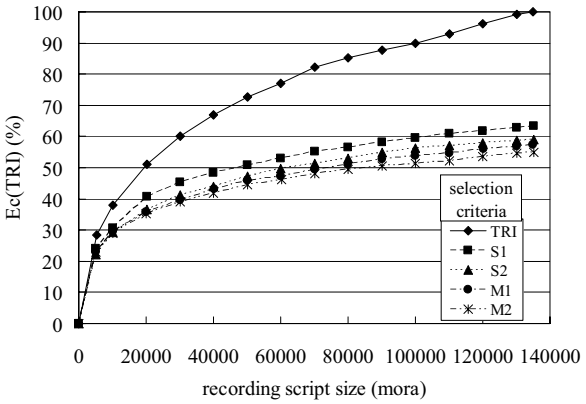

Figure 1. Ec(TRI) rates for each selection criterion as a function of recording script size.

Table 3. TCRs of expansion elements

selection criteria	TCR (%)			
	Tc(S1)	Tc(S2)	Tc(M1)	Tc(M2)
TRI	93.7	61.6	48.4	19.3
S1	95.6	63.7	50.2	20.5
S2	93.4	65.9	51.6	22.1
M1	92.8	64.0	54.9	24.1
M2	91.9	63.0	53.2	25.9

The sentence score of TRI is based on ECR. Because TRI is a basic element, it is necessary for the script to have all TRIs in the text corpus, regardless of frequency. Preliminary experiments showed that a smaller script can be generated by using ECR than by using TCR to collect all kinds of TRI in the text corpus. Table 2 shows the contents of the text corpus in this experiment.

Figure 1 shows the results of the experiment: the relationship between Ec(TRI) and selection criteria. It shows TRI can be collected completely using TRI criterion for text selection. However, TRI collection is insufficient if S1, S2, M1, and M2 criteria are used for text selection at about 140000 mora. It is necessary to give priority to TRI in sentence selection.

We also examined *Tc*(S1), *Tc*(S2), *Tc*(M1), and *Tc*(M2) using TRI or expansion elements as the selection criterion. The variations of the expansion elements are huge, we can not collect all of the variations. We must put priority on collecting high frequency elements. Therefore, the sentence scores of S1, S2, M1, and M2 are based on TCR. Table 3 shows that resulting script size reaches 140000 mora. It shows that S1, S2, M1, and M2 collection is weak if TRI is used for sentence selection.

3. PROPOSED RECORDING SCRIPT GENERATION ALGORITHM

Following the experiments in 2.3, we developed a script generation method based on a greedy algorithm with 2-stage selection. The 1st stage handles the basic element (TRI). The 2nd stage handles the expansion elements.

The algorithm is described as follows:

1st stage. The score of all sentences in the text corpus are calculated. This score, denoted by *s*(TRI), is defined as the increase in *Ec*(TRI) that would occur if the sentence were added to the script. If there is only one sentence which has the highest score, add this sentence to the recording script, delete the sentence from the text corpus, and iterate the 1st stage.

2nd stage. If there are multiple sentences with equally highest score by *s*(TRI), calculate new score *S* using expansion elements to decide the most suitable one of these sentences. This new sentence score *S* is calculated as follows:

$$S = w(\mathrm{S1})s(\mathrm{S1})+w(\mathrm{S2})s(\mathrm{S2})+w(\mathrm{M1})s(\mathrm{M1})+w(\mathrm{M2})s(\mathrm{M2}) \quad (3)$$

where, *s*(S1) is defined as the increase of *Tc*(S1), that would occur if the sentence were added to the script. *s*(S2),*s*(M1) and *s*(M2) are defined in the same way. *w*(S1),*w*(S2),*w*(M1) and *w*(M2) are weight coefficients of S1,S2,M1 and M2, respectively. If there is only one sentence with highest score, add this sentence to the recording script and delete the sentence from the text corpus. If there are multiple sentences with equally highest score, select the sentence with the shortest length. Return to the 1st stage.

This loop is iterated until the size of the script, *Tc*(*X*) and/or *Ec*(*X*) meets some application-specific requirements.

4. SCRIPT GENERATION EXPERIMENT

4.1 Conditions

To examine the 2-stage selection algorithm, script generation with six sets of weight coefficients was carried out. We compared TCRs. Termination conditions and text corpus are the same as in the experiment in 2.3. The details of six sets, (a)~(f), are as follows.

In weight coefficient set(a), we set all weights to 0. That is *w*(S1)=*w*(S2)=*w*(M1)=*w*(M2)=0: sentence score *S* in the 2nd stage was always 0. This was intended to examine TCRs if 2nd stage selection was not used.

Table 4. TCRs using 2-stage selection.

weight coefficient conditions	TCR (%)			
	Tc(S1)	Tc(S2)	Tc(M1)	Tc(M2)
(a) (single stage) *w*(S1)=*w*(S2)= *w*(M1)=*w*(M2)=0	93.7	61.6	48.4	19.3
(b) (2-stage) *w*(S1)=1, *w*(S2)=*w*(M1) =*w*(M2)=0	94.9	62.8	49.5	20.0
(c) (2-stage) *w*(S2)=1, *w*(S1)=*w*(M1) =*w*(M2)=0	93.9	64.0	50.3	20.8
(d) (2-stage) *w*(M1)=1, *w*(S1)=*w*(S2) =*w*(M2)=0	93.8	63.1	51.7	21.5
(e) (2-stage) *w*(M2)=1, *w*(S1)=*w*(S2) =*w*(M1)=0	93.7	63.0	51.0	22.2
(f) (2-stage) *w*(S1)=*w*(S2) =*w*(M1)=*w*(M2)=1	94.0	63.5	50.5	21.5

In weight coefficient sets(b)~(e), we set one of the weights to 1, the others to 0. For example, *w*(S1)=1, *w*(S2)=*w*(M1) =*w*(M2)=0. In weight coefficient set(f), we set all weights to 1. That is *w*(S1)=*w*(S2)=*w*(M1)=*w*(M2)=1. These sets were intended to examine which types of phonetic elements were included in the script via the weight coefficients of phonetic elements.

4.2 Results and discussion

Table 4 shows the weight coefficient sets and the resulting TCRs at the script size of 140000 mora. Comparing with the single stage method, set(a), the proposed 2-stage method, sets(b)~(f), obtains mostly higher TCR scores. It indicates that 2-stage selection is effective in raising TCR for all expansion elements.

The TCR with sets(b)~(e) show that each TCR has the highest value if its phonetic element and phonetic element for selection are the same. In the case of weight coefficient set(f), although the TCRs are not the highest, they can provide, on average, higher scores than the other weight coefficient sets. This indicates that taking account of several expansion elements can collect a wider variety of phonetic elements on average. This experiment shows that the proposed method yields scripts that contain a wide variety of phonetic elements while eliminating as many redundant phonetic elements as possible.

5. LISTENING EXPERIMENT

5.1 Conditions

Section 4 examined the proposed script generation method from the viewpoint of TCR. This section examines it from the

viewpoint of the subjective quality of synthesized speech. The following experiment was carried out to determine how much the expansion elements influence the synthesized speech quality; we assume a speech database for the specified task of newscasting.

First, two scripts were generated from the same text corpus: newscast genre. One script (script-S1) was generated using w(S1) = 1 with all other weight coefficients = 0. The other script (script-M1) was generated using w(M1) = 1 with all other weight coefficients = 0. A TTS speech database was created from each script. Each speech database had about 1 hour speech (including about 27000 mora). Database-S1 was created from script-S1, and database-M1 was created from script-M1.

A preference test was carried out by 5 subjects who were experienced in listening to synthesized speech. 10 sentences from a newscast task were synthesized by our corpus-based TTS system[10] using the two speech databases. The sampling rate of the synthesized speech was 22050Hz. These speech pairs were presented through headphones. In addition, another 10 sentences from mixed tasks other than newscast (for example novel, information guidance, and so on.) were synthesized and examined in the same way.

5.2 Results and discussion

The results are shown in Figure 2. For the newscast task, database-M1 yielded better speech quality than database-S1. In this task, the task of the text corpus and synthesized speech are the same. This indicates that M1 better matches the feature of the task than S1. This is reasonable because S1 is an acoustic element and M1 is a linguistic element.

For the other tasks, database-S1 speech was preferred over that of database-M1. In this case, the task of the text corpus and synthesized speech are different. This indicates that S1 is more task-independent than M1. In another way, script-S1 is more general-purpose than script-M1.

These results indicate two conclusions. First, which expansion element is used to generate a script influences the quality of the synthesized speech. Second, when we build a task-specific speech database, we should take account of morphemes when generating the script.

6. CONCLUSION

This paper introduced a new script generation method for constructing efficient speech databases for corpus-based TTS systems. We proposed an algorithm that can generate scripts containing a wide variety of speech elements, and that allows the types of elements included in the script to be controlled. Experimental results show our 2-stage selection process is effective in raising the text cover rate for all expansion elements. The proposed method yields scripts that contain a sufficiently wide variety of phonetic elements while simultaneously eliminating as many redundant phonetic elements as possible. A preference test showed that differences in the type of phonetic element for the script generation criteria influence the quality of the synthesized speech. The preference test also showed that M1 is better than S1 when generating a task-specific script. In the future, we will examine other weight coefficients of more phonetic element combinations. We will also examine the use of prosody in script generation.

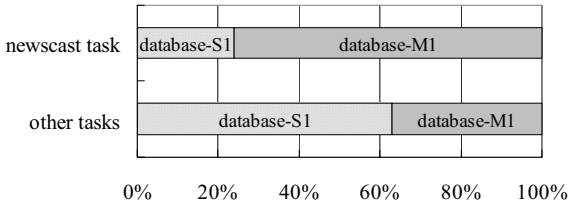

Figure 2. Results of preference tests.

ACKNOWLEDGMENTS

The authors would like to thank Hisashi Ohara, Akihiro Imamura, Masahiro Oku and Masanobu Abe for their guidance in this research. The authors are also grateful to the members of the Speech, Acoustics and Language Laboratory for their helpful discussions.

REFERENCES

[1] T. Hirokawa and K. Hakoda, "Segment Selection and Pitch Modification for High Quality Speech Synthesis using Waveform Segments, " *Proc. of ICSLP'90*, pp.337-340, 1990.

[2] N. Campbell, "CHATR: A High-Definition Speech Resequencing System, " Proc. of 3rd ASA/ASJ Joint Meeting, pp.1223-1228, 1996.

[3] M. Beutnagel, A. Conkie, J. Schroeter, Y. Stylianou, and A. Syrdal, "The AT&T Next-Gen TTS System," *Proc. of 137th Meeting Acoustic Society America*, 1999.

[4] K. Iso, T. Watanabe, and H. Kuwabara, "Design of a Japanese Sentence List for a Speech Database," *Proc. of Spring Meeting of The Acoustical Society of Japan,* pp.89-90, 1988, (*in Japanese*).

[5] J. P. H. van Santen, "Methods for Optimal Text Selection," *Proc. of Eurospeech97*, pp.553-556, 1997.

[6] H. Kawai, S. Yamamoto, and T. Shimizu, "A Design Methods of Speech Corpus for Text-to-Speech Synthesis Taking Account of Prosody," *Proc. of ICSLP 2000,* pp.420-425, 2000.

[7] C. Kuo and J. Huang., "Efficient and Scalable Methods for Text Script Generation in Corpus-based TTS Design, " *Proc. of ICSLP2002,* pp.121-124, 2002.

[8] B. Bozkurt, O. Ozturk, and T. Dutoit, "Text Design for TTS Speech Corpus Building Using a Modified Greedy Selection," *Proc. of Eurospeech 2003*, pp.277-280, 2003.

[9] G. Brassard and P. Bratley., "Fundamentals of Algorithmics, " Prentice-Hall, Inc. 1996

[10] K. Mano, H. Mizuno, H. Nakajima, H. Asano, M. Isogai, M. Hasebe, and A. Yoshida, "Development of a corpus-based concatenative text-to-speech synthesis system 'Cralinet' for contact center services, " *Proc. of Autumn Meeting of The Acoustical Society of Japan,* pp.347-348, 2004, (*in Japanese*).