The VoiceText Text-to-Speech System for the Blizzard Challenge 2007

Won-Suk Jun, Deok-Su Na, Seong-Won Kim, Myung Kim, Joon-Woo Lee and Jong-Seok Lee

Voiceware, Co., Ltd, Seoul, Korea
{jws, dsna, lagun, jinming, joonwoo, jslee}@voiceware.co.kr

Abstract

This paper introduces the VoiceText text-to-speech system developed by Voiceware. By means of corpus based concatenative speech synthesis technique, we built high quality synthetic voices using the dataset provided for the Blizzard challenge 2007. The evaluation results show that VoiceText achieved high performances in both naturalness and intelligibility of synthesized speech.

1. Introduction

The Blizzard challenge was devised in 2005 in order to better understand different speech synthesis techniques on a common dataset. An organized evaluation, based on listening tests, was carried out to try to rank the systems and help identify the effectiveness of the techniques [1]. We have been very interested in this challenge since the results of the Blizzard challenge 2005 were published. At last, we decided to participate in the third challenge this year in order to evaluate our VoiceText text-to-speech (TTS) system.

VoiceText is a multi-lingual TTS system developed by Voiceware Co., Ltd. It has been a very successful commercial TTS product since it was released in 2000. It supports high-quality speech synthesis for Korean, US-English, Japanese and Mandarin Chinese with various voices. Recently, US-Spanish TTS is under development.

Corpus based concatenative speech synthesis, which is one of the most widely used techniques to synthesize natural sounding speech, is adopted in VoiceText. This approach has the advantage of obtaining very high quality synthesized speech by carefully selecting and concatenating appropriate synthesis units contained in a large speech corpus. On the other hand, it suffers from large memory and high computational burden. VoiceText tackles this problem by incorporating efficient algorithms to reduce database size and computational complexity. In addition, the process of preparing the speech DB is mostly automated, allowing new voices to be built quickly with ease.

This paper describes a brief overview of VoiceText system and its application to the Blizzard challenge 2007.

2. Overview of VoiceText TTS System

VoiceText consists of four main modules, which are text processing, prosody generation, unit selection and speech signal processing, as shown in Figure 1.

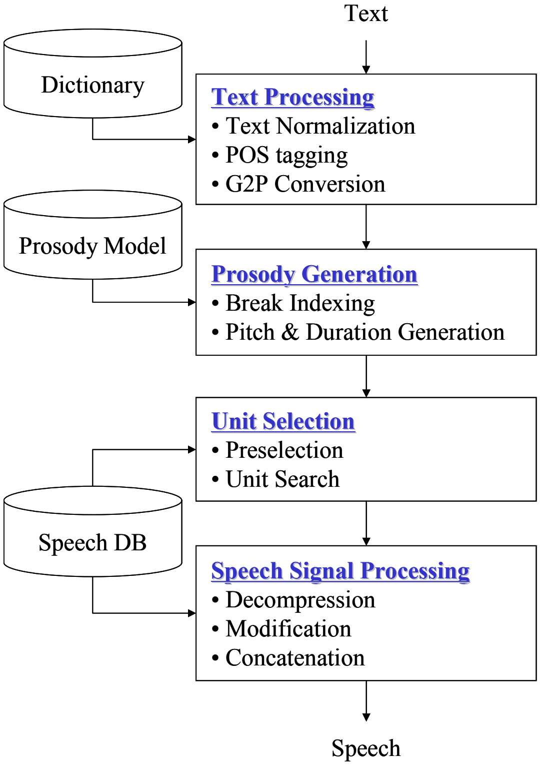

Figure 1: Block diagram of VoiceText TTS system.

2.1. Text processing

The text processing module is composed of text normalization, POS (Part of Speech) tagging and G2P (Grapheme-to-Phoneme) conversion.

Text normalization conducts such operations as finding end of sentence, tokenization of the input into words, abbreviation expansion and numeral expansion.

POS tagger determines the parts of speech corresponding to words that constitute a sentence. Due to the highly ambiguous characteristics of natural language, a word may have more than one POS, which is determined according to the context of the given sentence. VoiceText uses a probabilistic model to calculate the most appropriate POS based on the high-level context information.

G2P conversion is the process which generates word pronunciations using a dictionary or letter-to-sound rules.

2.2. Prosody generation

Prosody refers to the emphasis and intonation of speech intended to form a rhythm using pitch, duration, amplitude and pauses. In order to obtain natural synthetic speech, we need to effectively model and process natural prosodic information. Prosody is affected by factors such as phonetic context, sentence structure, semantics, and emotion. Currently, it is rather difficult to accurately analyze and process

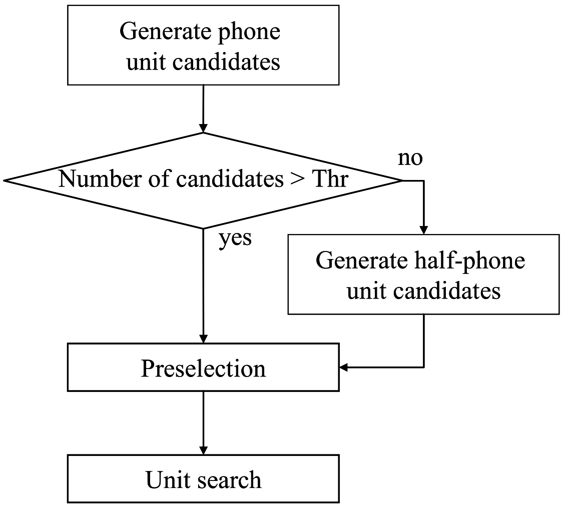
Generate phone unit candidates
Number of candidates > Thr
no
yes
Generate half-phone unit candidates
Preselection
Unit search

Figure 2: Block diagram of unit selection.

information pertaining to semantics and emotion, so we use a statistical method to model prosody with emphasis on sentence structure and phonetic context.

For prosody generation, a tree-based method known as CART (Classification and Regression Trees) [2, 3] is used. The structure of the CART algorithm is intuitive and easy to understand. Thus, it has the advantage that post-processing techniques can be devised with ease. And it can also minimize the efforts for supporting new voices and languages.

2.3. Unit selection

Since, in a large corpus based TTS engine, there exist many candidates corresponding to a unit to be synthesized, unit selection is most crucial to the overall quality of the TTS output. It is based on two cost functions, namely, target cost and concatenation cost. In our system, phonetic context, duration and pitch are used for target cost, and cepstrum, pitch and energy are used for concatenation cost. The unit selection procedure is the task of determining the optimal sequence of units so that the accumulated sum of these two costs is minimized [4]. It is generally performed with a dynamic programming (DP) search algorithm. To reduce computational complexity, a Viterbi beam search is used.

The half-phone based unit selection system increases the naturalness of synthesized speech significantly but results in high complexity [5]. To achieve both high quality and low computational complexity, we use phone units and half-phone units selectively.

Figure 2 shows the structure of our unit selection method. First, phone unit candidates are generated for a target phone. In case that the number of phone unit candidates is not more than a threshold (Thr), half-phone unit candidates are generated instead.

After the generation of unit candidates, preselection is conducted to reduce the number of unit candidates. In this process, simple and fast cost calculations are performed over all possible unit candidates, and the top *N* candidates are selected [6]. We use Connected Context Length (CCL) [7] and target cost as preselection costs. CCL is the length of the phonetic context sequence which of each unit candidate matches the target phonetic context sequence. This may mean how many consecutive unit candidates are actually connected in the speech database. After preselection, only survived unit candidates are involved in further complicated computations such as concatenation cost calculation and unit search process.

2.4. Speech signal processing

Speech compression techniques can be applied to reduce the size of the speech database. Especially, the compression approach is very efficient for the application of the TTS systems designed for hand-held devices having limitation on memory usage [8]. In case that the speech database is compressed, the decompression must accordingly be carried out to reconstruct speech waveforms. And some speech modifications can also be applied so as to change prosody parameters such as pitch, speed and volume. The final speech output is generated by just concatenating speech segments in a pitch synchronous way.

In the present paper, we used a high quality speech compression/decompression technique whose compression ratio is about 1/4. But no speech modification is conducted at all.

3. Voice building for the Blizzard challenge 2007

The database for the Blizzard challenge 2007 is composed of 6579 utterances recorded by a male speaker with an U.S. English accent. Total time length of speech is around 8 hours excluding the leading and tail silences in each waveform file. The provided data includes four kinds of files as belows.

- txt: transcription of the corresponding waveform.
- wav: 16 kHz sampling waveform in Microsoft RIFF format.
- utt: Festival Utterance format.
- lab: Xlabel format phone labels.

We utilized our own word pronunciations and voice building system across the whole data processing. All we need to build a new voice are only speech and text. Thus, ‘utt’ and ‘lab’ files were discarded and only ‘wav’ and ‘txt’ files were used.

3.1. Phonetic transcription

The phonetic transcriptions of utterances were generated fully automatically by the text processing module of VoiceText. There could be some mismatches between utterances and their phonetic transcription. They are caused by speaker’s reading mistakes, text normalization errors or G2P errors. To acquire the exact speech output and achieve the best speech synthesis quality, all these mismatches must be corrected. This is a very time-consuming work, but the timeline for the Blizzard Challenge 2007 does not allow participants to have enough time to inspect all transcriptions of 8-hour speech corpus thoroughly. So we checked transcriptions somewhat roughly. This task took us about 80 person-hours.

3.2. Phone segmentation

The phonetic transcriptions were automatically aligned with the corresponding speech by using speaker-dependent HMMs. The HMMs were trained using a small amount of phone-segmented data by VoiceEZ, which is the HMM-based speech recognition system developed by Voiceware. To generate initial phone-segmented data for the given speaker, we conducted some manual work, which is actually the task to correct the phone labels obtained from speaker-independent HMMs. It took us about 4 person-hours to process 15-minute speech data used in training.

After automatic generation of phone labels by HMM based alignment, no manual correction was carried out to refine the phoneme boundaries.

3.3. Building voices

Each participant is asked to build three synthetic voices from the database as belows.

- Voice A: from the full dataset (about 8 hours).
- Voice B: from the ARCTIC subset (about 1 hour).
- Voice C: from a subset of the data chosen by participant.

Building voice A and B is the same task as the previous challenge [9]. As for voice C, participants can choose utterances under the condition that the total duration of the selected utterances must be no more than the duration of the ARCTIC subset. The selection method should be based on only the text information.

Under these constraints, we devised a text selection method using a greedy algorithm to maximize the phonetic context coverage of selected texts. The process is very similar to the method which extracts phonetically rich sentences from a large text corpus [10].

We built run-time databases of three synthetic voices corresponding to the above datasets. The run-time database is composed of prosody models (pitch and duration model), compressed speech segment database and unit inventory which contains phonetic contexts, unit boundary parameters and speech segment information. All components of each synthetic voice were generated by making use of only the appropriate dataset.

3.4. Synthesizing speech

The Blizzard participants were asked to synthesize total 400 sentences from five different genres. We converted all sentences to synthesized speech waveform files using only VoiceText system without any help from other systems. Table 1 shows the ratio of total time length of synthesized speech files to total synthesis time when each system was run on a Xeon 3 GHz processor. VoiceText can synthesize speech over 250 times faster than the real time for all three voices. Although the dataset of Voice A (about 8-hour) is much bigger than the others (about 1-hour), there is no big difference in synthesis time thanks to the aggressive preselection strategy and the efficient unit search of our system.

4. Evaluation results

The Blizzard Challenge organizers conducted the evaluation based on four kinds of listening tests as follows.

(i) Similarity to original speaker: 5-point scale score from 1 [Sounds like a totally different person] to 5 [Sounds like exactly the same person].

(ii) Comparison of two samples in terms of their overall naturalness: the alternative of Similar or Different.

(iii) Mean Opinion Score (MOS): 5-point scale score from 1 [Completely Unnatural] to 5 [Completely Natural].

(iv) Semantically Unpredictable Sentence (SUS) test: Word Error Rate (WER).

The (i) and (ii) are new types of tests different from those of the previous Blizzard challenges. In this paper, we introduce some selected results of the evaluation tests. Detailed analyses and results of all kinds of tests will be presented by the Blizzard challenge organizers [11].

Table 2 shows evaluation results of our system for three voices. Our system demonstrates better performance with the full dataset (voice A) than with the significantly smaller dataset (voice B or C) as most corpus based speech synthesis

Table 1: Comparison of synthesis time ratios.

Voice	Total length of speech files (sec)	Total synthesis time (sec)	Ratio
A	1486	5.62	264
B	1425	5.28	270
C	1519	5.22	291

Table 2: Evaluation results of VoiceText system for three voices.

Voice	Similarity	MOS	WER (%)
A	3.7	3.6	24
B	3.4	3.3	32
C	3.2	3.1	37

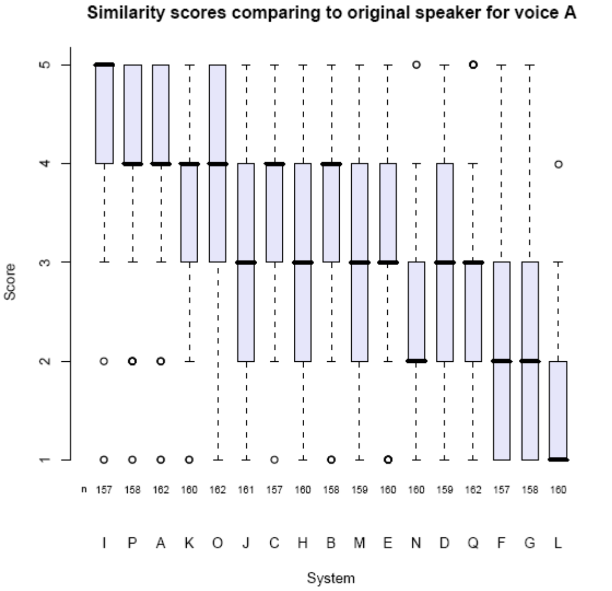

Figure 3: Box plot of similarity scores comparing to original speaker for voice A.

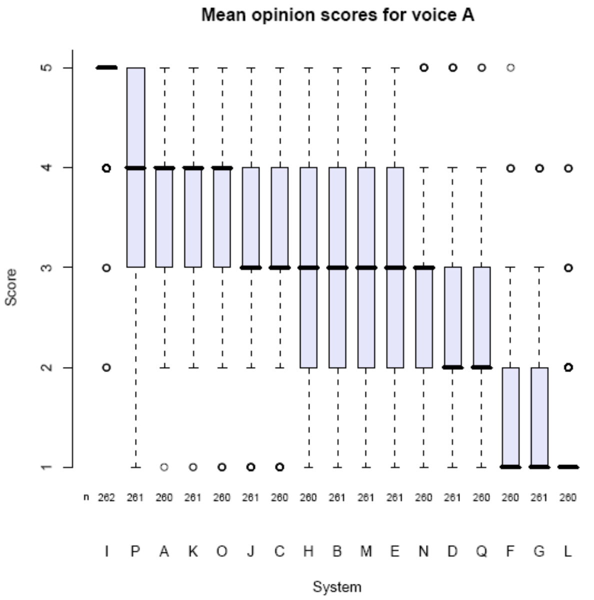

Figure 4: Box plot of mean opinion scores for voice A.

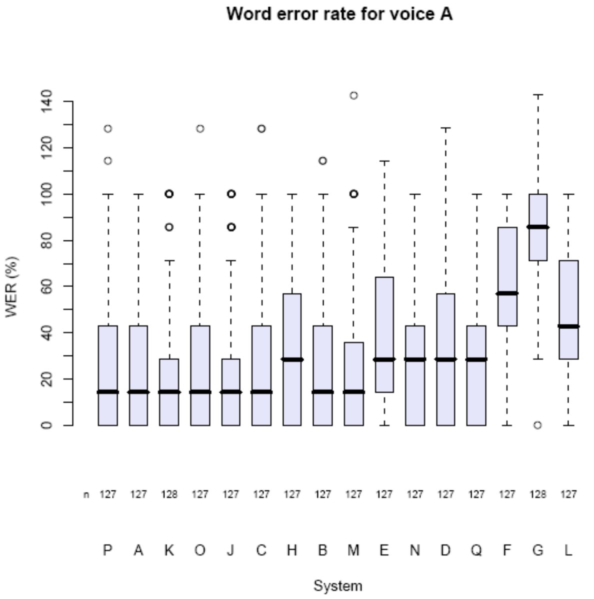

Figure 5: Box plot of word error rates for voice A.

systems do [9]. But the differences are not very large and voice B (or C) also performs fairly well in spite of using very small dataset.

Voice C was constructed from a subset of the data chosen by our own text selection algorithm described briefly in Section 3.3. Although we expected voice C to perform better than voice B, the result is unfortunately the opposite of our expectation. The text selection method to just maximize the phonetic context coverage of selected utterances seems not to be effective.

Figure 3, 4 and 5 show standard box plots [12] of similarity scores, mean opinion scores and word error rates for voice A. These display median (central solid bar), quartiles (shaded box), 1.5*quartile range (extended lines) and outliers (circles). In the figures, ‘n’ is the number of data, ‘K’ is our system and ‘I’ is natural speech. Our system achieved high performance near the top in every evaluation test although our fast unit selection algorithm and speech compression/decompression technique might slightly degrade speech quality.

5. Conclusions

This paper describes the VoiceText text-to-speech system for the Blizzard challenge 2007. Using efficient algorithms in unit selection, VoiceText can synthesize speech at very high speed. And it shows good performances at all types of evaluation tests such as similarity to original speaker, mean opinion score and word error rate.

In this challenge, a text selection algorithm was designed to find more optimal subset from the full dataset, but it caused some degradation in comparison with the ARCTIC subset. It seems that other factors than maximizing the phonetic context coverage must be considered to improve the performance.

6. References

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

[2] S. Lee and Y. H. Oh, “Tree-based modeling of prosodic phrasing and segmental duration for Korean TTS systems,” Speech Communication, vol. 28, pp. 283-300, 1999.

[3] S. Lee and Y. H. Oh, “Tree-based modeling of intonation,” Computer Speech & Language, vol. 15, pp. 75-98, January 2001.

[4] A. Hunt and A. Black, “Unit Selection in a Concatenative Speech Synthesis System Using a Large Speech Database,” Proc. of ICASSP, vol.1, pp.373-376, 1996.

[5] M. Beutnagel, A. Conkie, J. Schroeter, Y. Stylianou and A. Syrdal, “The AT&T Next-Gen TTS System,” Proc. of 137th Meeting of the ASA, 1999.

[6] A. Conkie, M. Beutnagel, A. Syrdal, and E. Brown, “Preselection of Candidate Units in a Unit Selection-Based Text-to-Speech Synthesis System,” Proc. of ICSLP, vol.3, pp.314-317, 2000.

[7] D. S. Na, W. S. Jeon, J. W. Lee, M. H. Cho, J. S. Lee, and M. J. Bae, “A Pre-selection Method Using Accentual Phrase Matching in Unit Selection-Based Japanese Text to Speech,” Proc. of WESPAC IX, pp.124-130, 2006.

[8] C. H. Lee, S. K. Jung, T. Eriksson, W. S. Jun, and H. G. Kang, “An Efficient Segment-Based Speech Compression Technique for Hand-Held TTS Systems,” Proc. of Interspeech, pp. 213-216, 2006.

[9] C. L. Bennett and A. W. Black, “Blizzard Challenge 2006: Results,” Proc. of Blizzard Challenge 2006 workshop, Sept. 2006.

[10] J. Ni, T. Hirai, and H. Kawai, “Constructing a phonetic-rich speech corpus while controlling time-dependent voice quality variability for English speech synthesis,” Proc. of ICASSP, pp. 881–884, 2006.

[11] http://festvox.org/blizzard/

[12] http://en.wikipedia.org/wiki/Box_plot/