I²R Text-to-Speech System for Blizzard Challenge 2011

Minghui Dong, S W Lee, Paul Chan, Ling Cen

Institute for Infocomm Research (I²R), Agency for Science, Technology and Research, Singapore 138632

{mhdong, swylee, ychan, lcen}@i2r.a-star.edu.sg

Abstract

This paper describes I²R's submission to the Blizzard Challenge 2011 speech synthesis evaluation. This is our fourth participation in the challenge. In this paper, we will describe our main approaches to building the required voices. We will describe our definitions of the acoustic, prosodic and linguistic parameters, procedure of candidate unit selection, components of cost functions, etc. Finally, we will also present the results of the listening test conducted.

Index Terms: speech synthesis, HMM-based synthesis, unit selection, and cost function.

1. Introduction

The Blizzard Challenge [1-3] provides an excellent platform for speech synthesis researchers to evaluate one another's corpus-based speech synthesis technology using the same database. This year, only English speech synthesis systems are evaluated.

There are two tasks in this year's Blizzard Challenge, namely hub task EH1 and spoke task ES1. Both of them are for English and use databases by the same speaker, 'Nancy'. This speaker is a female professional speaker with US accent. About 12,000 utterances from this speaker are available. These utterances are provided in a sample rate of 16 kHz. Apart from these files, the original studio recording files with higher sampling rates are also available.

A set of annotation data were also provided with the training data in the Lesseme format. Lessemes are symbolic sound representations derived from Lessac's phonosensory symbols, linking guidelines and intonational information. Hence, co-articulation and prosodic features are notated. During the recording, the Lesseme annotation is shown to the speaker. Hence, sound-annotation correspondence may be achieved.

In EH1, entry systems are required to generate synthetic speech on novel, news, reportorial, and semantically unpredictable sentences (SUS). In ES1, it is required to synthesize sentences comprising of names and addresses.

2. Overview of Our Approach

Over the years, unit-selection based waveform concatenation [4] and hidden Markov model (HMM) based parametric synthesis [5] are two popular approaches for corpus-based speech synthesis. Although the unit-selection approach often provides natural and high quality synthesis outputs, there may be artifacts during waveform concatenation on small databases. Compared to the unit-selection approach, only a small amount of training data is required for the HMM based parametric approach to generate synthesis outputs of acceptable quality. Nevertheless, since the HMM based approach relies on source-filter modeling for output generation, the synthesized speech sounds vocoded and robotic.

This year, the I²R entry adopts unit-selection based approach as our major method for synthesis. However, we employ the HMM-based parametric synthesis approach to assist in the unit-selection process, so as to capture the high quality and smooth continuity in real speech with smooth trajectory. This is similar to [6, 7]. In particular, based on the parameter trajectory generated from the HMM-based synthesis system, the inventory of waveform candidates is evaluated and an optimal subset of candidates is chosen.

In the following sections, we will first introduce the prosody model, the candidate set selection method, and the unit-selection process. Then we will look at the evaluation result, and finally draw the conclusion.

3. Prosody Model

We have used the same prosody models as we did last year [8]. In this part, we describe how the prosody model of the speech synthesis system was built.

3.1. The Acoustic Parameters

We first calculated a set of parameters that describe the spectral and prosodic features of each HMM state as well as frame boundaries. These parameters are chosen to include all the possible parameters in our consideration. The main values that we capture include the statistical values of each individual HMM state as well as the values of boundary (start and end) frames of the unit. The initial parameter set that we used consists of the following values:

- Spectral features: MFCC mean for the 3 HMM states, MFCC for boundary frames.
- Pitch features: Mean, maximum, minimum, and range of pitch values and pitch derivative values for 3 HMM states, and boundary frames.
- Duration features: Durations of the 3 states, duration of the unit.
- Energy features: Mean energy of frames in the 3 HMM states, and boundary frames.

The defined parameter set forms a long vector (with a dimension of 308), which contains a lot of redundancy. Therefore, we use the principal component analysis approach to reduce the dimension. The dimensionally-reduced vector is considered a compact form of representation of the prosodic and spectral features of the unit. Finally, we have a 40-dimensional vector.

3.2. The Prosodic Parameters

The acoustic parameters define both spectral and prosodic information. However, because more parameters are required

to convey spectral information as compared to prosodic information, prosodic information is actually less prominent in the acoustic vector. Nevertheless, we still need a set of prosodic parameters to emphasize the prosodic properties in speech. The prosodic parameters for each unit consist of the following:

- Pitch mean of the unit
- Duration of the unit
- Energy mean of the unit
- Pitch range of the unit.

3.3. Linguistic Features

Linguistic features are derived from the input text. They are used for predicting the acoustic parameters. Even though the training data come along with a set of annotation in the form of Lessemes, the annotation was not used in our system due to the difficulty to make it compatible with some of our components. We used Festival to analyze the text and generate the utterance structure for each speech file. The defined linguistic feature set is similar to that used in the HTS system [9].

We have derived the following linguistic features from the utterance files (the number of parameters are given in brackets):

- Current and context units: phone identities of current unit, the previous 2 and next 2 units, phone positions (counting forward and backward) in the syllable. (7)
- Syllable information: Stress, accent, length of the previous, current and next syllables. (9)
- Syllable position information: syllable position in word and phrase, stressed syllable position in phrase, accented syllable position in phrase, distance from the stressed syllable, distance from the accented syllable, and name of the vowel in the syllable. (13)
- Word information: length and part-of-speech of the previous word, current word and next word, position of the word in phrase. (12).
- Phrase information: Lengths (in number of words and syllables) of previous phrase, current phrase and next phrase, position of the current phrase in major phrase, boundary tone of the current phase. (9)
- Utterance information: Lengths in number of syllables, words and phrases. (3)

Putting all the features together, we form an input linguistic feature vector of 53 elements.

3.4. Parameter Prediction

The acoustic parameter prediction process calculates the parameters from the linguistic features. The prediction can be represented with the following formula:

$$y_i = F_i(X) \tag{1}$$

where y_i is the i-th parameter for the unit and X is the linguistic feature vector for the unit.

In our system, the linguistic features are the predictors and the acoustic and prosodic parameters are the responses. We build our models using the CART [10] approach. Each individual parameter is predicted separately with a CART tree.

4. Candidate Set Selection

During synthesis, there is a large set of candidate units in the unit database for each target unit. Therefore, to improve the synthesis speed, it is important to reduce the candidate set first. In our system, HMM-based synthesis method was used to determine a candidate set for each unit.

4.1. Generation of HMM Parameter Trajectory

Our HMM-based synthesis system is based on the HTS speaker-dependent training demo released in [11]. There are 12,095 sentences of training data available for the hub task EH1. To quickly train a HMM-based synthesis system, we use the first 1,000 sentences as our training set.

Feature extraction is first done to capture the spectrum, fundamental frequency (F0) and aperiodicity information. STRAIGHT [12] is used for analysis purpose. The resultant STRAIGHT spectrum is then converted to 39-th order line spectral pair (LSP) and combined with log gain and five-band aperiodicity values. These are the static features. LSP is employed for its efficiency for quantization and stability after interpolation [13]. Dynamic features, i.e. the delta and the delta-delta features are also used. These feature values are modeled by Multi-space probability distribution HMM (MSD-HMM) [14]. The feature vector finally consists of 138 dimensions, split into five streams, one stream for spectrum, another stream for band aperiodicity and three streams for log F0. The frame shift is 5 ms.

The training process follows the standard maximum-likelihood procedure as in [11]. Duration models and full-context decision trees are built. Finally, the parameters of the leaf nodes in the tree structures are extracted for candidate unit set selection.

4.2. Selection of Waveform Candidates

Based on the generated speech parameter sequences, we select a subset of waveform candidates for subsequent unit selection in the baseline synthesis system. This significantly reduces the computational cost, rather than searching the entire inventory. Given a text input, the generated sequence from HMM synthesis acts as the target unit. On the other hand, the entire inventory is divided into a number of subsets defined by the HMM clusters obtained from the HMM synthesis system. Each cluster collects waveform candidates having similar segmental and prosodic context. By calculating the Kullback-Leibler divergence (KLD) [15] between HMMs of every possible pair of the target unit and these clusters, the nearest subset of waveforms is determined. This is done by finding the waveform cluster with minimum KLD. Note that only models comprising spectrum, log F0 and band aperiodicity information are used for KLD calculation. Duration models are not involved. There are altogether 776 clusters found.

Let λ_p and λ_q be the HMMs of the target unit and one cluster, respectively. For our case, λ_p and λ_q are single-mixture Gaussian distributions, consisting of *S* states. For simplicity, it is assumed that all transitional probabilities between model states are the same. The KLD becomes

$$KLD(\lambda_p, \lambda_q) = \sum_{s=1}^{S} \frac{1}{2}[(D(\lambda_{p,s}, \lambda_{q,s}) + D(\lambda_{q,s}, \lambda_{p,s})] \tag{2}$$

$$D(\lambda_m,\lambda_n)=-\frac{1}{2}[\ln\frac{|\Sigma_m|}{|\Sigma_n|}-tr(\Sigma_m\Sigma_n^{-1})+N \quad (3)$$
$$-(\mu_m-\mu_n)'\Sigma_n^{-1}(\mu_m-\mu_n)]$$

where $\lambda_{p,s}$ denotes state s of λ_p. $|\cdot|$ and tr are the matrix determinant and the trace operation respectively. N is the number of dimensions of λ_p and λ_q. μ_m and Σ_m are the mean vector and covariance matrix of λ_m respectively.

During synthesis, the above KLD calculation is performed for every target unit in the input text. In order to speed up, the KLD values between all possible cluster pairs are computed in advance and stored. Hence, any KLD between a target unit and a cluster can be looked up and the subset of waveform candidates is easily retrieved.

5. Unit Selection

The unit-selection method is used in all the voices that we have built. In this section, we describe how we define the cost function.

The unit-selection process is based on the cost function that consists of two parts (1) a target cost to measure the difference between the target unit and the candidate unit. (2) a join cost to measure the acoustic smoothness between the concatenated units.

Our target cost further consists of three parts (1) the cost of acoustic parameters, (2) the cost of prosodic parameters, and (3) the cost of context linguistic features. The target cost c_t is defined as the following:

$$c_t = w_{ta}c_{ta} + w_{tp}c_{tp} + w_{tl}c_{tl} \quad (4)$$

where, c_{ta}, c_{tp} and c_{tl} are the cost of acoustic parameters, prosodic parameters and linguistic features respectively, and w_{ta}, w_{tp} and w_{tl} represent their corresponding weights.

The reason why we use three cost components here is that each of them alone is not sufficient to describe the target cost. The cost of the linguistic feature is to ensure the general spectral and prosodic accuracy of the candidate unit. Units with wrong pronunciation labels, which are generated due to grapheme-to-pronunciation mistakes, can also be excluded by linguistic cost. However, due to the varsity of speech, using linguistic cost on its own may lead to extreme cases of abnormal spectrum and prosody too easily. The use of cost of acoustic parameters can avoid the selection of such extreme cases, because statistical models favor average values. The use of prosodic cost is to emphasize the importance of prosodic features.

The total cost c is calculated with the following function.

$$c = w_t\sum_{i=0}^{n}c_t(i) + w_j\sum_{i=1}^{n}c_j(i) \quad (5)$$

where n is number of units in the sequence, $c_t(i)$ is the target cost of unit i, $c_j(i)$ is the join cost between unit i-1 and unit i, and w_t and w_j are weights for the target cost and join cost respectively.

The best unit sequence is determined by searching for a best path among the candidate unit lattice to minimize the total cost of the selected sequence. Viterbi algorithm is used to find the best sequence. The weights in the cost function are manually tuned.

6. Evaluation Results

The following presents some of the major evaluation results for our system in this year's Blizzard Challenge tasks. Our system is denoted as 'K', whereas system A, B, C and D are benchmark systems. System A refers to the real speech. System B is the Festival unit-selection system. System C and D are the HTS systems using similar techniques such as the HTS entry in Blizzard Challenge 2005 with 16 kHz and 48 kHz sampling rates respectively.

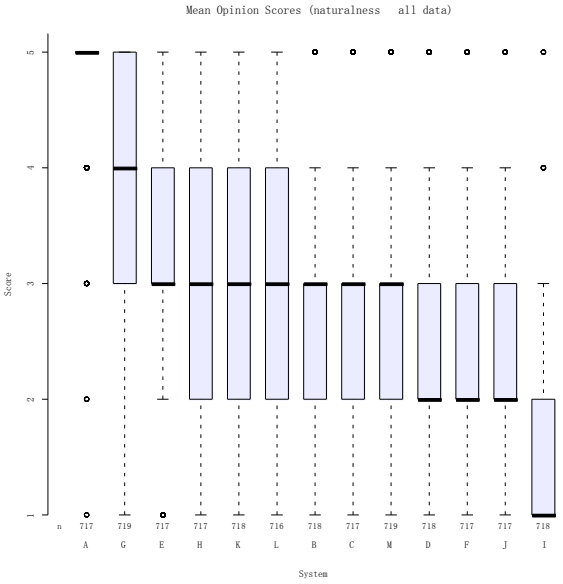

Figure 1: *MOS for naturalness of all data types.*

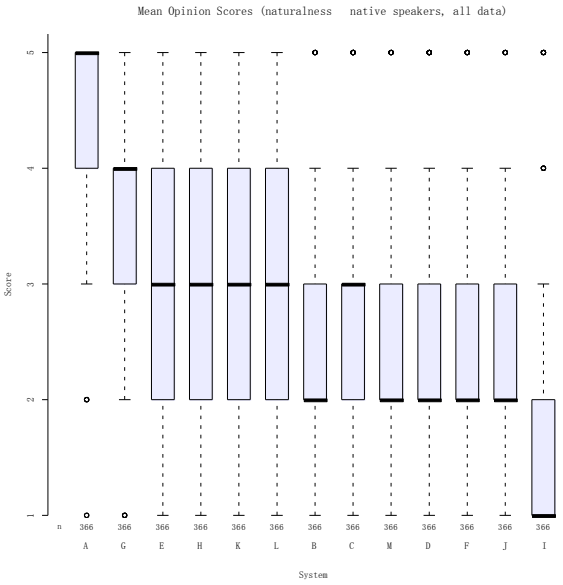

Figure 2: *MOS for naturalness of all data types, collected from native speakers.*

Concerning the naturalness, Fig. 1 shows the performance of our system for all data types. With our unit-selection synthesis system which is assisted by HMM-generated trajectory, the median of mean opinion score (MOS) is 3. This indicates that our system K generates output speech with high levels of naturalness. Compared to the three benchmark

systems (B, C and D), our system K shows significantly better performance, with reference to the Wilcoxon's signed rank tests.

Based on the MOS results on novel, news and reportorial testing texts (with all medians equal 3), it is found that the performance of our system is roughly the same across different speech types.

The naturalness of our system outputs is evaluated by both native and non-native English speakers. Fig. 2 and 3 are the MOS results. The result from native speakers is not as good as the result from non-native speakers. This indicates that there are probably some detailed characteristics in the output that are not natural enough, where only native speakers are able to identify.

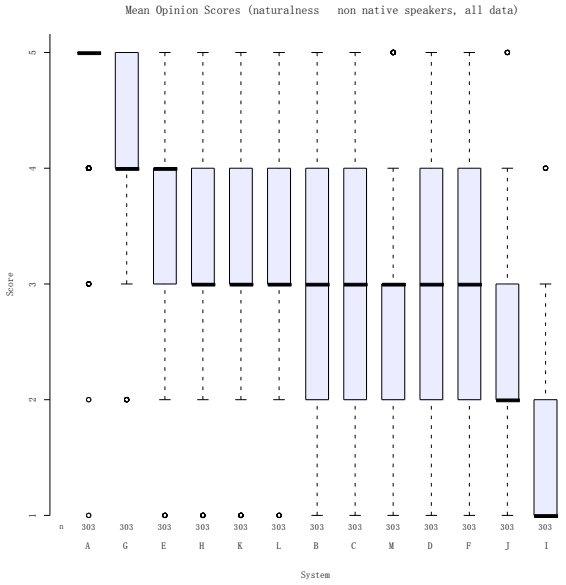

Figure 3: *MOS for naturalness of all data types, collected from non-native speakers.*

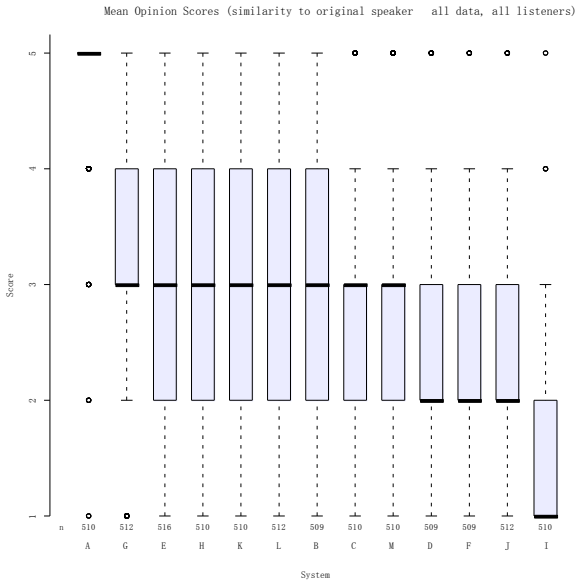

Figure 4: *MOS for similarity of all data types.*

Fig. 4 shows the similarity performance. This is for all data types and all listeners. The median of the MOS for our system K is 3. This indicates that our system preserves and generates the original speaker's characteristics well. According to the Wilcoxon's signed rank tests, our system is found to have a similarity performance paralleled to the benchmark system B from Festival.

If different speech types are compared, it is found that our system performs slightly better in capturing the original speaker's similarity for the news database as compared to the novel one.

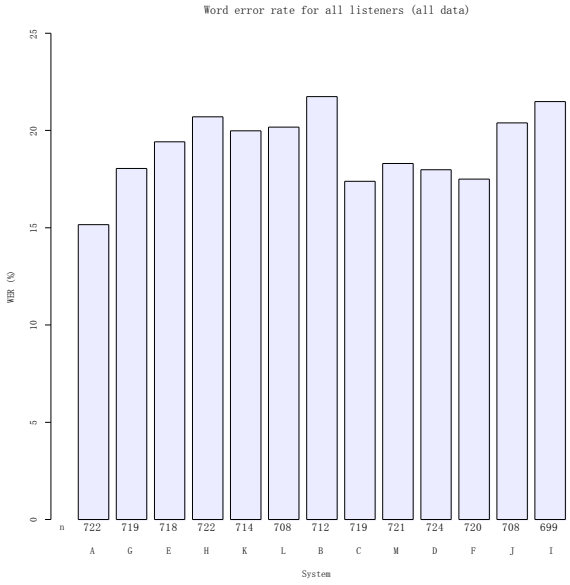

Figure 5: WER *of all data types and all listeners.*

Fig. 5 presents the system performances on intelligibility in the form of the error rates (WER) for all data types and all listeners. The measured WER under different categories are roughly the same, being in the middle-to-high range among all the systems.

Comparing our system with benchmark systems B, C and D, it is found that the two HMM-based systems (C and D) generally perform better in intelligibility, over system B and K, which are unit-selection based.

From the evaluation results, we noticed that our system works almost equally well for different types of text. There are still some aspects to improve: (1) We used Festival's text analysis component of the default voice as our front end. The analysis result contains some pronunciation errors for some words. We believe that improving the text analysis part will help to improve the overall speech quality of our system further. (2) Weights in cost functions are still manually tuned. Due to time constraints, the tunings were yet to be optimized. (3) We have used the HMM-based synthesis method to assist us in unit selection. We believe that we need more tests to take optimal advantage of this in our system.

7. Conclusion

This paper has described our speech synthesis approach for the Blizzard Challenge 2011. We have used the HMM-based approach to select the candidate unit set in a unit-selection based system. The evaluation shows that the performance of our system is equally good with different testing sets in terms of both naturalness and similarity.

References

[1] R. Clark, M. Podsiadlo, M. Fraser, C. Mayo, S. King, Statistical Analysis of the Blizzard Challenge 2007 Listening Test Results, Proc. Blizzard Challenge Workshop, 2007, Bonn, Germany.

[2] S. King, V. Karaiskos, "The blizzard Challenge 2009." Blizzard Challenge Workshop, Sept. 2009.

[3] A. W. Black and K. Tokuda, "The Blizzard Challenge - 2005: Evaluating corpus-based speech synthesis on common datasets," in Proc Interspeech 2005, Lisbon, 2005.

[4] A. W. Black and P. Taylor, "Automatically clustering similar units for unit selection in speech synthesis," Proc. Eurospeech, pp. 601-604, Sep. 1997.

[5] K. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi, and T. Kitamura, "Speech parameter generation algorithms for HMM-based speech synthesis," in Proc. ICASSP, pp. 1315-1318, Jun. 2000.

[6] Y. Jiang, Z.-H. Ling, M. Lei, C.-C. Wang, L. Heng, Y. Hu, L.-R. Dai, and R.-H. Wang, "The USTC system for Blizzard Challenge 2010," in Proc. Blizzard Challenge Workshop, Sep. 2010.

[7] Y. Qian, Z.-J. Yan, Y.-J. Wu, F. K. Soong, G. Zhang, and L. Wang, "An HMM trajectory tiling (HTT) approach to high quality TTS – Microsoft entry to Blizzard Challenge 2010," in Proc. Blizzard Challenge Workshop, Sep. 2010.

[8] M. Dong, P. Chan, L Cen, B. Ma, H. Li, "I2R Text-to-Speech System for Blizzard Challenge 2010", Blizzard Challenge Workshop, Sept. 2010.

[9] K. Tokuda, H. Zen, A.W. Black, "An HMM-based Speech Synthesis System Applied to English," in Proc. of 2002 IEEE SSW, Sept. 2002.

[10] L. Breiman, , J. H. Friedman, R. A. Olshen, and C. J. Stone, "Classification and Regression Trees". Monterey, Calif., U.S.A.: Wadsworth, Inc., 1984.

[11] K. Tokuda, K. Oura, K. Hashimoto, S. Shiota, H. Zen, J. Yamagishi, T. Toda, T. Nose, S. Sako, and A. W. Black, (2011, Jul. 28) HMM-based speech synthesis system (HTS) [Online]. Available: http://hts.sp.nitech.ac.jp/

[12] H. Kawahara, (2011, Jul. 28) STRAIGHT trial page. [Online]. Available: http://www.wakayama-u.ac.jp/~kawahara/STRAIGHTtrial/

[13] F. K. Soong and B.-H. Juang, "Line spectrum pair (LSP) and speech data compression," in Proc. ICASSP, pp. 37-40, Mar. 1984.

[14] K. Tokuda, T. Mausko, N. Miyazaki, and T. Kobayashi, "Multi-space probability distribution HMM," IEICE Trans. Inf. & Syst., vol. E85-D, pp. 455-464, Mar. 2002.

[15] S. K. Zhou and R. Chellappa, "Kullback-Leibler distance between two Gaussian densities in reproducing kernel Hilbert space," in Proc. ISIT, Jun. 2004.