The USTC and iFlytek Speech Synthesis Systems for Blizzard Challenge 2007

Zhen-Hua Ling[1], *Long Qin*[1], *Heng Lu*[1], *Yu Gao*[1], *Li-Rong Dai*[1], *Ren-Hua Wang*[1], *Yuan Jiang*[2], *Zhi-Wei Zhao*[2], *Jin-Hui Yang*[2], *Jie Chen*[2], *Guo-Ping Hu*[2]

[1]University of Science and Technology of China
[2]iFlytek Research, Hefei, Anhui, China
zhling@ustc.edu

Abstract

This paper introduces the speech synthesis systems developed by USTC and iFlytek for Blizzard Challenge 2007. These two systems are both HMM-based ones and employ similar training algorithms, where contextual dependent HMMs for spectrum, F0 and duration are estimated according to the acoustic features and contextual information of training database. However, different synthesis methods are adopted for these two systems. In USTC system, speech parameters are generated directly from these statistical models and parametric synthesizer is used to reconstruct speech waveform. The iFlytek system is a waveform concatenation one, which uses maximum likelihood criterion of statistical models to guide the selection of phone-sized candidate units. Comparing the evaluation results of these two systems in Blizzard Challenge 2007, we find that the parametric synthesis system achieves better performance than unit selection method in intelligibility. On the other hand, the synthesized speech of the unit selection system is more similar to the original speech and more natural especially when the full training set is used.

1. Introduction

In recent years, HMM-based parametric speech synthesis method has been proposed and made significant progress [1-3]. In this method, spectrum, pitch and duration are modeled simultaneously in a unified framework of HMMs [1] and the parameters are generated from HMMs under maximum likelihood criterion by using dynamic features [4]. Then parametric synthesizer is used to reconstruct speech signals. This method is able to synthesize highly intelligible and smooth speech. Besides, the voice character of synthetic speech can be controlled flexibly by employing some model adaptation methods [5]. However the speech quality of this method suffers from the unnatural output of parametric synthesizer even if some high quality speech vocoder, such as STRAIGHT [6], has been used.

In order to overcome this problem, a HMM-based unit selection and waveform concatenation speech synthesis method has also been proposed [7,8]. In this method, likelihood and Kullback-Leibler divergence criterions of the trained HMMs are followed to select the optimal frame-sized or phone-sized unit sequence. Then the waveform of each candidate unit is concatenated to produce synthesized speech. The advantage of this method over conventional unit selection method is that statistical criterions are introduced into the calculation of target cost and concatenation cost, so the synthesis system can be trained automatically with little expert knowledge and manual tuning.

Two systems which adopt each of the HMM-based parametric synthesis method and unit selection method are developed by USTC and iFlytek for Blizzard Challenge 2007. The flowchart of these two systems is shown in Figure 1. They share almost the same training algorithms but are distinct from each other in synthesis stage.

This paper is organized as follows. Section 2 introduces the details about the HMM-based parametric synthesis system developed by USTC. Section 3 describes the unit selection method used in iFlytek system. Some descriptions about system building are presented in section 4. Section 5 gives the evaluation results and some discussions. Section 6 is the conclusion.

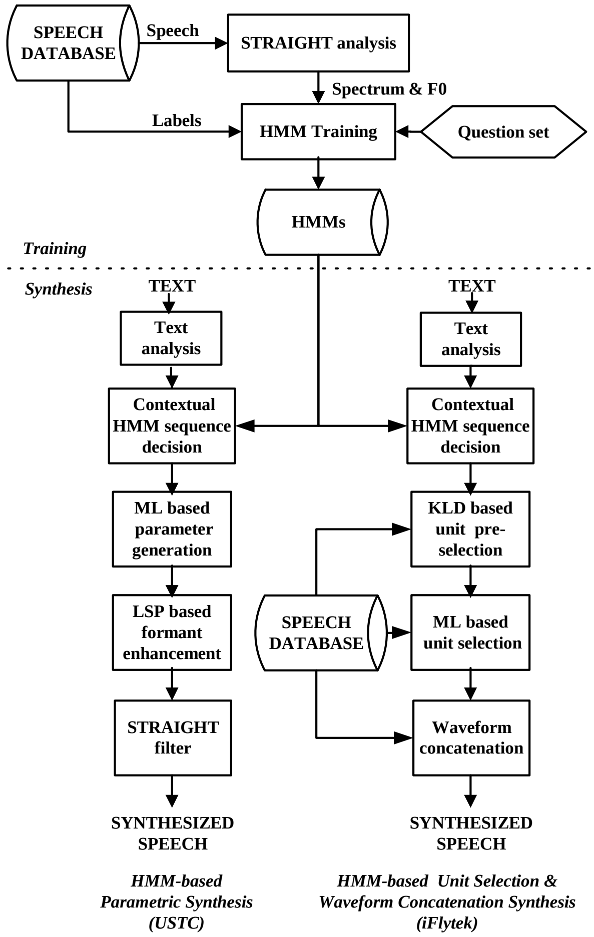

Figure 1: *Flowcharts of USTC and iFlytek systems for Blizzard Challenge 2007.*

2. HMM-based Parametric Synthesis System of USTC

2.1. Model training

At first, acoustic features are extracted from the speech waveforms of training database. STRAIGHT [6] as a high quality speech vocoder is adopted here to analyze the spectral envelop and F0 for each speech frame. In our system linear spectral pair (LSP) with frequency warping is selected to present each frame's spectrum considering LSPs relate more closely to formant positions and have better smoothness among adjacent frames [3]. The final feature vector for each frame consists of static, delta and delta-delta components of LSPs and logarithmized F0.

A set of contextual dependent HMMs are estimated according to the acoustic features and label information of the training database under maximum likelihood criterion [1]. The spectrum part is modeled by a continuous probability distribution and the F0 part is modeled by a multi-space probability distribution (MSD) [9]. A decision tree based model clustering method is applied after contextual dependent HMM training to improve the robustness of estimated models. During training, the state transition probability matrices for all contextual dependent HMMs with the same monophone label are tied. Then each utterance in the training database is segmented into states by Viterbi alignment using trained acoustic HMMs. Based on the results of state segmentation, contextual dependent state duration model and phone duration model are trained with the same decision tree clustering method as acoustic model training.

In order to improve the quality of synthesized speech, Minimum Generation Error (MGE) training [10] is carried out to update the model parameters given by maximum likelihood training. Under MGE criterion, the model parameters are estimated to minimize the difference between generated parameters and natural ones for the sentences in training set. The MGE criterion gives better consistency between model training and the purpose of speech synthesis, which is to produce speech signal or parameter sequences as closely as the natural ones. Besides, by incorporating parameter generation into the training procedure, the constraints between static and dynamic features are considered in HMM training. Here, only the model parameters of spectral part are updated by employing Generalized Probabilistic Descent (GPD) algorithms [11].

2.2. Parameter generation and speech synthesis

For an input text, the contextual dependent HMM sequences of whole sentence are determined by the clustered HMMs and decision trees according to the results of text analysis. The first step of parameter generation is to predict duration for each state in the sentence. Here, we combine the state duration model and phone duration model to make the prediction [3]. Then spectral and F0 parameters are generated by maximum likelihood parameter generation algorithm [4].

Because of the averaging effect of statistic modeling, the spectrums reconstructed from ML based parameter generation algorithm are always over-smoothed and the formants are broaden, which make the synthetic speech sounds muffled. Here, we modify the positions of generation LSPs for each frame to enhance the formants of synthesized speech considering the relationship between spectral peaks and LSP, especially the difference between its adjacent orders [3]. At last, the spectral envelop of each frame is recovered from the modified LSPs and sent into STRAIGHT synthesizer with F0 to generate final speech waveform.

3. HMM-based Unit Selection and Waveform Concatenation System of iFlytek

3.1. Model training

The model training for HMM-based unit selection is almost the same as the training processes introduced in section 2.1 except two differences.

1) Mel-cepstrums instead of warped LSPs are adopted as spectral features. Because here the acoustic features are used for unit selection, not speech reconstruction. So the details of spectrum are less cared and mel-cepstrums can give better description about the overall shape of spectral envelop with less feature orders. For mel-cepstrums, only ML training is carried out and MGE training is skipped.
2) Besides the acoustic model and duration model, a concatenation model is also trained to model the transition of acoustic features at phone boundaries. The feature of concatenation model is defined as the differential of mel-cepstrum and F0 between the first frame of current phone and the last frame of previous phone after state segmentation using trained acoustic model. In the same way, contextual dependent models are trained and decision tree based model clustering is applied.

Phone is used as the base unit for selection in this system. The segmentation of candidate phone units is realized automatically after the Viteibi alignment for the training of duration model and concatenation model.

3.2. ML-based unit selection

In this system, maximum likelihood criterion is employed to guide the selection of phone-sized candidate units. The optimal phone sequence is expected to be searched out from the speech database to maximize the combined likelihood of acoustic model, phone duration model and concatenation model.

Assuming the number of phones in the utterance for synthesis is N. For phone $n\,(n=1,...,N)$, the contextual dependent acoustic model, phone duration model and concatenation model determined by clustered HMMs and decision trees are λ_n, λ_n^{dur} and λ_n^{con}. For a whole sentence, the acoustic model, phone duration model and concatenation model sequences are written as λ, λ^{dur} and λ^{con}. One candidate unit for phone n is and the corresponding acoustic model of candidate unit u_n is λ_n^c. $\boldsymbol{o}_n = \{\boldsymbol{o}_{n,1},...,\boldsymbol{o}_{n,T_n}\}$ presents the acoustic feature vectors of unit u_n which consist of static and dynamic features for each frame. The dynamic features of current frame are calculated using the static features of previous, current and next frames [7]. For a whole utterance, the phone candidate sequence can be written as $\boldsymbol{u} = \{u_1,...,u_N\}$ and the optimal one $\boldsymbol{u}^*$ is determined using Eq.(1),

$$\boldsymbol{u}^* = \arg\max_{\boldsymbol{u}}[LL_{cmp}(\boldsymbol{u},\lambda) + LL_{dur}(\boldsymbol{u},\lambda^{dur}) + LL_{con}(\boldsymbol{u},\lambda^{con})] \tag{1}$$

where

$$LL_{cmp}(\boldsymbol{u},\lambda) = \sum_{n=1}^{N}\log P(\boldsymbol{o}_n \mid \lambda_n, Q_n) \tag{2}$$

$$LL_{dur}(\boldsymbol{u},\lambda^{dur}) = \sum_{n=1}^{N}\log P(T_n \mid \lambda_n^{dur}) \tag{3}$$

$$LL_{con}(\boldsymbol{u},\lambda^{con}) = \sum_{n=2}^{N}\log P(\boldsymbol{o}_{n,1} - \boldsymbol{o}_{n-1,T_{n-1}} \mid \lambda_n^{con}) \tag{4}$$

Here $LL_{cmp}(\boldsymbol{u},\lambda)$, $LL_{dur}(\boldsymbol{u},\lambda^{dur})$ and $LL_{con}(\boldsymbol{u},\lambda^{con})$ measure the likelihood of unit sequence $\boldsymbol{u}$ towards the sentence HMMs. Ignoring the influence of state transition probability and assuming that the state allocation Q_n for unit u_n is the same as the alignment between u_n and λ_n^c which is given by segmentation in training stage, Eq.(1) can be rewritten as Eq.(5) with some weights for different models

$$\boldsymbol{u}^* = \arg\min_{\boldsymbol{u}}\left\{\sum_{n=1}^{N}[W_{cmp}\bullet\frac{T_n^p}{T_n}\bullet MD(\boldsymbol{o}_{n,i},\boldsymbol{m}_{n,i},\Sigma_{n,i}) + W_{dur}\bullet MD(T_n, m_n^{dur}, \sigma_n^{dur\,2})] + \sum_{n=2}^{N}W_{con}\bullet MD(\boldsymbol{o}_{n,1}-\boldsymbol{o}_{n-1,T_{n-1}},\boldsymbol{m}_n^{con},\Sigma_n^{con})\right\} \tag{5}$$

$$MD(\boldsymbol{o},\boldsymbol{m},\Sigma) = (\boldsymbol{o}-\boldsymbol{m})^T\Sigma^{-1}(\boldsymbol{o}-\boldsymbol{m}) \tag{6}$$

where the likelihood of acoustic model is normalized by the candidate phone duration T_n and predict phone duration T_n^p; $\boldsymbol{m}_{n,i}$ and $\boldsymbol{\Sigma}_{n,i}$ are the mean vector and covariance matrix for the observation Gaussian PDF of frame i in u_n decided by λ_n and Q_n ; $\lambda_n^{dur} = \mathcal{N}(m_n^{dur},\sigma_n^{dur\,2})$; $\lambda_n^{con} = \mathcal{N}(\boldsymbol{m}_n^{con},\Sigma^{con})$; W_{cmp}, W_{dur} and W_{con} are some weights that are set manually; $MD(\bullet)$ is the Mahalanobis distance function. In order to facilitate unit search process, Eq.(5) can be converted to the traditional form of a sum of "target cost" and "concatenation cost" as Eq.(7) ,

$$\boldsymbol{u}^* = \arg\min_{\boldsymbol{u}}\{\sum_{n=1}^{N}TC(u_n) + \sum_{n=2}^{N}CC(u_{n-1},u_n)\} \tag{7}$$

where $TC(u_n)$ and $CC(u_{n-1},u_n)$ denote the target cost of unit u_n and the concatenation cost of units u_{n-1} and u_n respectively, given as

$$TC(u_n) = W_{cmp}\bullet\frac{T_n^p}{T_n}\bullet\sum_{i=2}^{T_n-1}MD(\boldsymbol{o}_{n.i},\boldsymbol{m}_{n,i},\Sigma_{n.i}) + W_{dur}\bullet MD(T_n,m_n^{dur},\sigma_n^{dur\,2}) \tag{8}$$

$$CC(u_{n-1},u_n) = W_{cmp}\bullet\frac{T_n^p}{T_n}\bullet MD(\boldsymbol{o}_{n,1},\boldsymbol{m}_{n,1},\Sigma_{n,1}) + W_{cmp}\bullet\frac{T_{n-1}^p}{T_{n-1}}\bullet MD(\boldsymbol{o}_{n-1,T_{n-1}},\boldsymbol{m}_{n-1,T_{n-1}},\Sigma_{n-1,T_{n-1}}) + W_{con}\bullet MD(\boldsymbol{o}_{n,1}-\boldsymbol{o}_{n-1,T_{n-1}},\boldsymbol{m}_n^{con},\Sigma_n^{con}) \tag{9}$$

Dynamic programming search can be realized using Eq.(7)~(9). Compared with conventional definition of target cost and concatenation cost, these costs given here are derived automatically and little manual designing and tuning is necessary.

3.3. KLD-based unit pre-selection

In order to reduce the computation cost of dynamic programming search, a Kullback-Leibler divergence based unit pre-selection algorithm is carried out. Here, we measure the KLD between the HMM of target unit and the HMM of each candidate unit to select the *K*-best units with minimum KLD before the calculation of target cost. However, for two HMMs there is no closed form solution for calculating the KLD between them. One alternative way is to estimate it by sampling using Monte-Carlo methods, but it will lead to very high complexity. Here, the upper bound of KLD between two left-to-right HMMs [12] is adopted as Eq.(10).

$$KLD(\lambda,\tilde{\lambda}) \le \sum_{i=1}^{S}\{\frac{D(\mathcal{N}(\boldsymbol{m}_i,\Sigma_i)\,||\,\mathcal{N}(\tilde{\boldsymbol{m}}_i,\tilde{\Sigma}_i))}{1-a_{ii}} + \frac{D(\mathcal{N}(\tilde{\boldsymbol{m}}_i,\tilde{\Sigma}_i)\,||\,\mathcal{N}(\boldsymbol{m}_i,\Sigma_i))}{1-\tilde{a}_{ii}} + \frac{(a_{ii}-\tilde{a}_{ii})\log(a_{ii}/\tilde{a}_{ii})}{(1-a_{ii})(1-\tilde{a}_{ii})}\} \tag{10}$$

where S is the number of states in a model; $\mathcal{N}(\boldsymbol{m}_i,\Sigma_i)$ and $\mathcal{N}(\tilde{\boldsymbol{m}}_i,\tilde{\Sigma}_i)$ present the observation PDF of state i for model λ and $\tilde{\lambda}$; a_{ii} and $\tilde{a}_{ii}$ present the state transition probability for λ and $\tilde{\lambda}$. Because λ and $\tilde{\lambda}$ must present the same monophone in our system and the transition probability matrix is tied, $a_{ii} = \tilde{a}_{ii}$ and Eq.(10) can be simplified as

$$KLD(\lambda,\tilde{\lambda}) \le \sum_{i=1}^{S}\frac{1}{1-a_{ii}}\{D(\mathcal{N}(\boldsymbol{m}_i,\Sigma_i)\,||\,\mathcal{N}(\tilde{\boldsymbol{m}}_i,\tilde{\Sigma}_i)) + D(\mathcal{N}(\tilde{\boldsymbol{m}}_i,\tilde{\Sigma}_i)\,||\,\mathcal{N}(\boldsymbol{m}_i,\Sigma_i))\} \tag{11}$$

For each state, the KLD between two *D*-dimension single mixture Gaussian distributions can be calculated as [13]

$$D(\mathcal{N}(\boldsymbol{m}_i,\Sigma_i)\,||\,\mathcal{N}(\tilde{\boldsymbol{m}}_i,\tilde{\Sigma}_i)) = \frac{1}{2}\ln(\frac{|\tilde{\Sigma}_i|}{|\Sigma_i|}) - \frac{D}{2} + \frac{1}{2}tr(\tilde{\Sigma}_i^{-1}\Sigma_i) + \frac{1}{2}(\tilde{\boldsymbol{m}}_i-\boldsymbol{m}_i)^T\tilde{\Sigma}_i^{-1}(\tilde{\boldsymbol{m}}_i-\boldsymbol{m}_i) \tag{12}$$

Because the state observation PDFs of all contextual dependent HMMs are clustered using decision tree in our system, Eq.(12) can be calculated offline as a matrix for every two leaf nodes in the decision tree of each state before synthesis. Therefore the unit pre-selection step can be realized efficiently.

3.4. Waveform concatenation

At last, the waveforms of every two consecutive candidate unit in the optimal phone sequence $\boldsymbol{u}^*$ are concatenated to produce synthesized speech. The cross-fade technique [14] is used here to smooth the phase discontinuity at concatenation points.

4. System Building

4.1. Speech database

The speech database for Blizzard Challenge 2007 contains 6579 utterances of about 8 hours. 3 voices are required to submit. For voice A, the full training set is used; voice B is built with only the ARCTIC subset, which contains 1032 utterances; voice C is constructed using a designed subset of the full training database, which is required to follow some restrictions. A greedy search algorithm [15] is used to select the utterance for system C building. At last 835 sentences are selected and the total duration is 2901.38 seconds. The result of corpus design for system C is shared by both USTC system and iFlytek system.

4.2. Implementation

Acoustic features were extracted at 5ms frame shift during STRAIGHT analysis and 5-state left-to-right without skip HMM structure was used in the model training. In USTC system, the order of LSP analysis was set to 40; 20 iterations were taken for GPD algorithms in MGE training. In iFlytek system, the order of mel-cespstrum was 13 (including 0-order) and the best 50 units were kept after KLD-based unit pre-selection; W_{cmp} was set to 1/39 for spectral part and 1/3 for F0 part; W_{dur} and W_{con} were set to 25 and 2 respectively.

5. Evaluation

The evaluation results of Blizzard Challenge 2007 for these two systems are discussed in this section. In following figures and tables, the labels of USTC system and iFlytek system are "J" and "A". System "I" denotes the natural speech.

5.1. Similarity test

The boxplots [16] of similarity scores of all systems for voice A, B and C are shown in Figure 2, 3 and 4. From these figures we can see that the synthesized speech of system A is more similar to the original speech than system J for all three voices. This can be attributed to the influence of parametric synthesizer, which causes muffled speech quality and degrades the similarity of synthesized speech.

5.2. Mean opinion score test

The boxplots of mean opinion scores of all systems for voice A, B and C are shown in Figure 5, 6 and 7. Tabel 1 gives the results of Wilcoxon's signed rank tests between the two proposed systems and other systems. It can be found from these figures and table that:

1) System A achieves better naturalness than system J in all three voices. This difference is significant only for voice A, which uses full training set. For the other two voices, the difference is not significant. This is because that the unit selection and waveform concatenation method gains more improvement than the parametric synthesis method when the size of the database increases.
2) System A is one of the best systems for all of the three voices. This proves the effectiveness of the HMM-based unit selection method which is realized automatically with little manual tuning. Besides, performance of system J is also competitive among all systems for voice B and C.

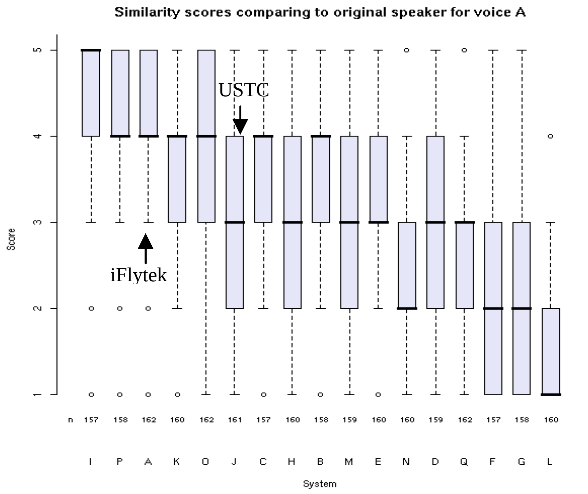

Figure 2: *Boxplot of similarity scores for voice A. The median (central solid bar), quartiles (shaded box), 1.5*quartile range (extended lines) and outliers (circles) of each system are displayed.*

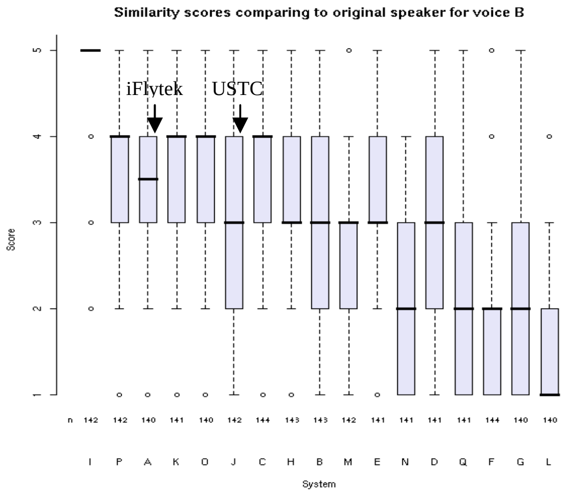

Figure 3: *Boxplot of similarity scores for voice B*

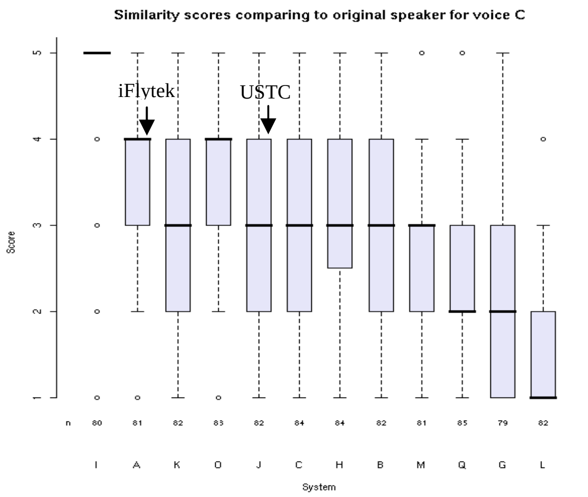

Figure 4: *Boxplot of similarity scores for voice C*

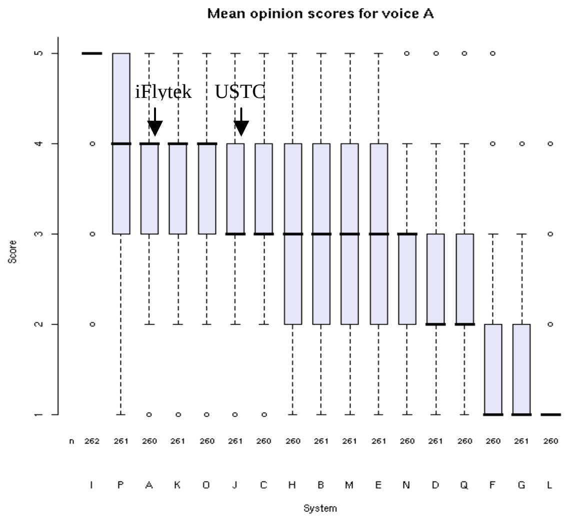

Figure 5: *Boxplot of mean opinion scores for voice A*

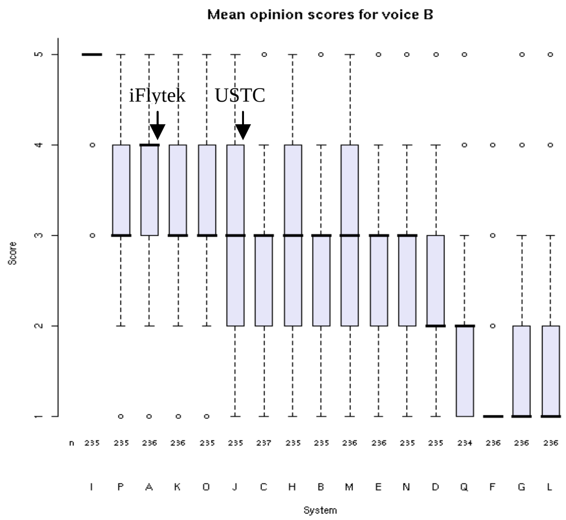

Figure 6: *Boxplot of mean opinion scores for voice B*

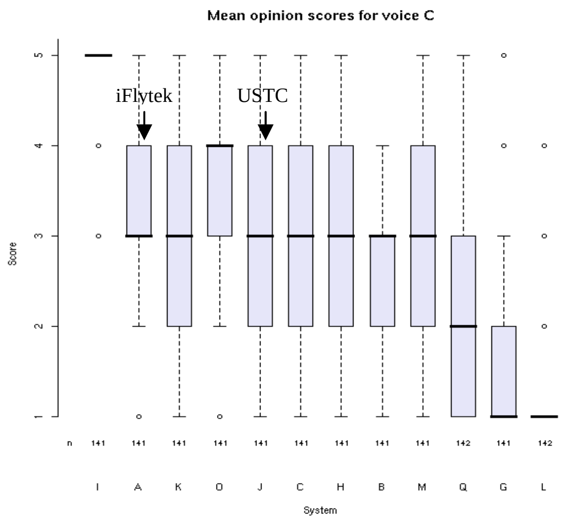

Figure 7: *Boxplot of mean opinion scores for voice C*

Table 1: Results of Wilcoxon's signed rank tests between system A, J and other systems (1 – significantly different; 0 – not significantly different)

	Voice A		Voice B		Voice C	
	A	**J**	**A**	**J**	**A**	**J**
P	0	1	0	0		
A (iFlytek)		1		0		0
K	0	0	0	0	0	0
O	0	0	0	0	0	0
J (USTC)	1		0		0	
C	1	0	1	1	1	0
H	1	0	1	0	1	0
B	1	1	1	1	1	1
M	1	1	1	1	1	0
E	1	1	1	1		
N	1	1	1	1		
D	1	1	1	1		
Q	1	1	1	1	1	1
F	1	1	1	1		
G	1	1	1	1	1	1
L	1	1	1	1	1	1

5.3. Word error rate test

Figure 8, 9 and 10 draw the results of word error rate test of all systems. Here system J shows its superiority over system A in mean WER for all of the three voices. So the intelligibility may be viewed as an advantage of parametric synthesis method over unit selection and waveform concatenation method because the synthesized speech of parametric synthesis system is more robust especially for the semantically unpredictable sentences used in WER test and the MGE training improves the intelligibility performance further. However, the WER differences between these two systems are not significant and they both have the statistically equal lowest WER for all three voices.

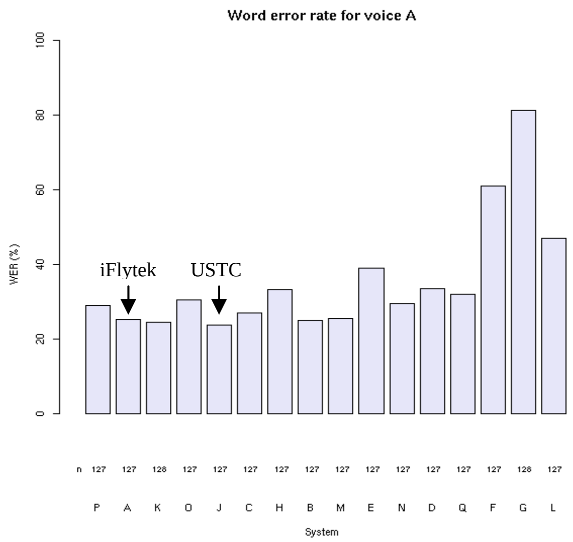

Figure 8: *Mean WER for voice A*

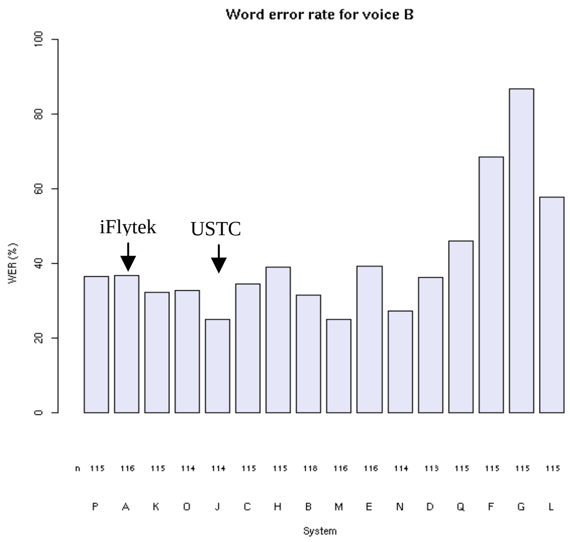

Figure 9: *Mean WER for voice B*

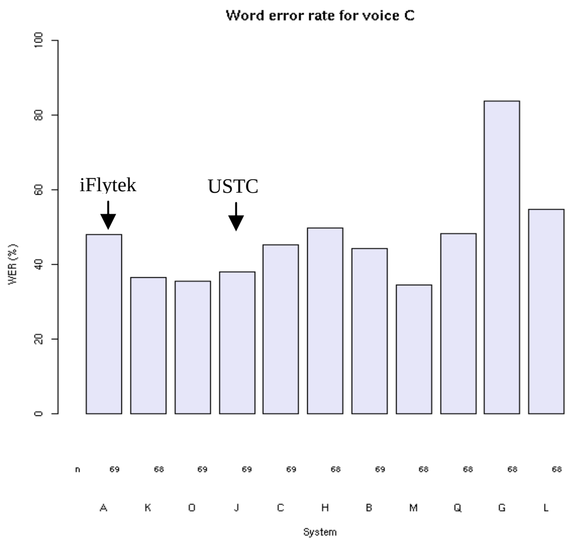

Figure 10: *Mean WER for voice C*

6. Conclusions

This paper introduces two HMM-based speech synthesis systems for Blizzard Challenge 2007. The USTC system adopts parametric synthesis method and the iFlytek system follows unit selection and waveform concatenation approach. After similar HMM training algorithm, maximum likelihood criterion is adopted in both systems no matter for parameter generation or for unit selection. The evaluation results show the different advantage of these two systems. The intelligibility of USTC system is better while the similarity and MOS scores of iFlytek system are higher. With the increasing of training data, the superiority of iFlytek system in naturalness becomes significant.

7. Acknowledgements

This work was partially supported by Hi-Tech Research and Development Program of China (Grant No.: 2006AA01Z137, 2006AA010104) and National Natural Science Foundation of China (Grand No.: 60475015).

8. References

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

[2] H. Zen, and T. Toda, "An Overview of Nitech HMM-based Speech Synthesis System for Blizzard Challenge 2005", in *Proc. of Eurospeech*, pp. 93-96, 2005.

[3] Zhenhua Ling, Yijian Wu, Yuping Wang, Long Qin, and Renhua Wang, "USTC system for Blizzard Challenge 2006 an improved HMM-based speech synthesis method", in *ICSLP Satellite Workshop, Blizzard Challenge*, 2006.

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

[5] M. Tamura, T. Masuko, K. Tokuda, and T. Kobayashi, "Adaptation of pitch and spectrum for HMM-based speech synthesis Using MLLR", in *Proc. of ICASSP*, pp. 805-808, 2001.

[6] H.Kawahara, I. Masuda-Katsuse, and A. de Cheveigne, "Restructuring speech representations using pitch-adaptive timefrequency smoothing and an instanta-neous-frequency-based F0 extraction: possible role of a repetitive structure in sounds," *Speech Communication*, vol. 27, pp. 187-207, 1999.

[7] Zhenhua Ling, and Renhua Wang, "HMM-based unit selection using frame sized speech segments", in *Proc. of ICSLP*, 2006, pp. 2034-2037.

[8] Zhenhua Ling, and Renhua Wang, "HMM-based Hierarchical Unit Selection Combining Kullback-Leibler Divergence with Likelihood Criterion", in *Proc. of ICASSP*, 2007, pp. 1245-1248.

[9] K. Tokuda, T. Masuko, N. Miyazaki, and T. Kobayashi, "Hidden Markov models based on multi-space probability distribution for pitch pattern modeling", in *Proc. of ICASSP*, pp. 229-232, 1999.

[10] Yijian Wu, and Renhua Wang, "Minimum generation error training for HMM-based speech synthesis", in *Proc. of ICASSP*, pp. 89-92, 2006.

[11] J. R. Blum, "Multidimensional stochastic approximation methods," *Ann. Math. Stat*, vol. 25, pp.737-744, 1954.

[12] Peng Liu, and Frank K. Soong, "Kullback-Leibler Divergence between Two Hidden Markov Models", *Microsoft Research Asia, Technical Report*, 2005.

[13] Christopher Rozell, "Information-Theoretic Analysis of Neural Responses in the Frequency Domain", *Rice University, Techinical Report*, 2003.

[14] T. Hirai, and S. Tenpaku, "Using 5 ms segments in concatenative speech synthesis," in *Proc. of 5th ISCA Speech Synthesis Workshop*, Pittsburgh, PA, USA, pp. 37-42, 2004,

[15] Ling Sun, Yu Hu, and Renhua Wang, "Corpus Design for Chinese Speech Synthesis System", in *Proc. of 6th National Conference on Man-Machine Speech Communication*, pp. 313-317, 2001. [in Chinese]

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