Internal Use Only（非公開）

TR－SLT－0068

HMM-based application for speech synthesis
音声合成へのHMM技術の応用

Yi-Jian Wu　Hisashi Kawai　Jinfu Ni　Renhua Wang
呉　義堅　河井　恒　倪　晋富　王　仁華

2004年3月16日

概要

This report includes three major parts. The first part is HMM-based segmentation by combining the minimum-segmentation-error based discriminative training and explicit duration modeling techniques. The second part is HMM-based prosody modeling for Chinese speech synthesis application. The contextual feature and the question set are designed according to the Chinese characteristics. Also, we improve the tree-based clustering by considering the space weight and the meaning of questions. The last part is automatic detection of Japanese vowel devoicing for corpus construction. The implied likelihood differences are extracted from the recognition process as the voicing measure. Also, we apply the discriminative training for voiced/devoiced HMM training, and incorporate the voicing features, including autocorrelation, energy and duration, to improve the detection performance.

（株）国際電気通信基礎技術研究所
音声言語コミュニケーション研究所
〒619－0288「けいはんな学研都市」光台二丁目2番地2 TEL：0774－95－1301

Advanced Telecommunication Research Institute International
Spoken Language Translation Research Laboratories
2-2-2 Hikaridai "Keihanna Science City" 619-0288,Japan
Telephone:+81-774-95-1301
Fax　:+81-774-95-1308

Abstract

My work includes three major parts. The first part is HMM-based segmentation by combining the minimum-segmentation-error based discriminative training and explicit duration modeling techniques. The second part is HMM-based prosody modeling for Chinese speech synthesis application. The contextual feature and the question set are designed according to the Chinese characteristics. Also, we improve the tree-based clustering by considering the space weight and the meaning of questions. The last part is automatic detection of Japanese vowel devoicing for corpus construction. The implied likelihood differences are extracted from the recognition process as the voicing measure. Also, we apply the discriminative training for voiced/devoiced HMM training, and incorporate the voicing features, including autocorrelation, energy and duration, to improve the detection performance.

I. HMM-based automatic segmentation

1 Minimum segmentation error based discriminative training

1.1 Background

In the corpus-based speech synthesis, the HMM-based automatic segmentation method had been popularly used for corpus construction. The conventional HMM training is based on Maximum Likelihood Estimation (MLE) criteria (via a powerful training algorithm, Expectation Maximization algorithm). In other words, this training method links the segmentation task to the problem of distribution estimation, and the HMMs are built to identify the phonetic segments, not to detect the boundary between the phonetic segments. This kind of inconsistency between the training and the application of HMM limits the segmentation performance.

In recent years, The discriminative training method and the criteria of Minimum Classification Error (MCE) based on the Generalized Probabilistic Descent (GPD) framework has been successful in training HMM for speech recognition [5][6], and to a certain extent segmentation can be regarded as a state recognition task with known transcription. This prompts us to apply the discriminative training method and the corresponding criteria for the segmentation task. Here, a new criteria, called minimum segmentation error (MSGE), is proposed to train the HMM under the GPD framework for the segmentation task.

1.2 Generalized Probabilistic Descent

Here is a brief introduction of the core part of GPD algorithm. For a given loss function $\ell(X,\Lambda)$, where X is a feature vector and Λ represents the system parameters, we want to optimize Λ to minimize the overall expectation loss:

$$L(\Lambda) = E[\ell(X,\Lambda)] = \int \ell(X,\Lambda) p(X) dX \,, \tag{I-1}$$

where $p(X)$ is *a priori* distribution. Since we do not know the *a priori* distribution, we cannot evaluate the expected loss directly. The Generalized Probabilistic Descent (GPD) algorithm[4] is a very powerful algorithm that can be used to accomplish this task. In a GPD framework, the target loss function is minimized according to an iterative procedure

$$\Lambda_{t+1} = \Lambda_t - \varepsilon_t U_t \nabla \ell(X_t,\Lambda)\Big|_{\Lambda=\Lambda_t} \,, \tag{I-2}$$

where U_t is a positive definite matrix, X_t is the tth training sample used in the sequential training process, and ε_t is a sequence of positive numbers that satisfies the conditions:

$$i)\sum_{t=1}^{\infty} \varepsilon_t \to \infty \quad \text{and} \quad ii)\sum_{t=1}^{\infty} \varepsilon_t^2 < \infty \,. \tag{I-3}$$

In the above, an infinite number of training samples is required for convergence. In practice, only a finite number of samples are available. However, we can minimize the empirical loss

$$L_0(\Lambda) = \frac{1}{N}\sum_{i=1}^{N}\ell(X_i,\Lambda) = \int \ell(X,\Lambda)p_N(X)dX \tag{I-4}$$

under the GPD framework. With sufficient training samples, the empirical loss converges to the actual expected loss. It should be noted that the GPD framework is a general framework for various definitions of loss function. A more detailed introduction and discussion of GPD algorithm can be found in the literatures [4][6].

1.3 Measurement for segmentation error

The conventional measurement of segmentation error is usually defined as the time difference in boundary location between human labeling and automatic labeling, i.e. *error length*. According to this definition, the segmentation errors are discrete (in frame scale) and not explicitly related to the parameters of the HMM. Therefore, the gradient-based optimization methods cannot be used to minimize the segmentation errors directly. We should find another suitable measurement for the segmentation errors.

Usually, the HMM-based segmentation is a state alignment procedure performed by the Dynamic Programming algorithm (e.g. Viterbi). For simplification, we look into the segmentation procedure of a sample X that consists of two connected segment units X_1 and X_2, i.e. $X = \{X_1, X_2\}$. In the DP algorithm, the likelihood of the best state alignment is calculated by

$$g_b(X;\Lambda) = \max_{Q} g(X,Q;\Lambda) = g(X,\overline{Q}_b;\Lambda), \tag{I-5}$$

where $\overline{Q}_b$ is the optimal state sequence with maximum likelihood, which is calculated as

$$g(X,\overline{Q};\Lambda) = \log P(X,\overline{Q};\Lambda) = \sum_{t=1}^{T}[\log a_{\overline{q}_{t-1}\overline{q}_t} + \log b_{q_t}(x_t)] + \log \pi_{\overline{q}_0}, \tag{I-6}$$

where $a_{\overline{q}_{t-1}\overline{q}_t}$ and $b_{q_t}(x_t)$ are transition probability and output probability distribution, respectively.

With the optimal state alignment, the corresponding phonetic boundary is labeled at time t', which satisfies the condition that $\overline{q}_{t'-1}$ is the final state of first unit and $\overline{q}_{t'}$ is the first state of the next unit. If the boundary is not the same as the humanly labeled boundary, i.e. the correct boundary, the optimal state alignment is regarded as "incorrect" state alignment. Also, the "correct" state alignment is defined as the optimal state alignment with the correct phonetic boundary restriction, which satisfies

$$g_c(X;\Lambda) = g_1(X_1,\overline{Q}_{c1};\Lambda) + g_2(X_2,\overline{Q}_{c2};\Lambda) = g(X,\overline{Q}_c;\Lambda) \tag{I-7}$$

where $\overline{Q}_{c1}$ and $\overline{Q}_{c2}$ are respectively the optimal state sequences of X_1 and X_2, and $\overline{Q}_c = \{\overline{Q}_{c1}, \overline{Q}_{c2}\}$.

Accordingly, we defined *error degree* as the difference in likelihood between the incorrect and the correct state sequence, i.e.

$$E_d = g_b(X,\Lambda) - g_c(X,\Lambda). \tag{I-8}$$

where $g_b(X,\Lambda)$ and $g_c(X,\Lambda)$ are the likelihood of incorrect and correct state sequences, respectively. When the segmentation is correct, i.e. $\overline{Q}_b = \overline{Q}_c$, E_d is equal to 0. If E_d is larger than 0, this indicates that the segmentation is incorrect and the value of E_d reflects how large the segmentation error is in some aspect. In order to find the meaning of error degree in depth, we analyzed the correlation between error degree and error length.

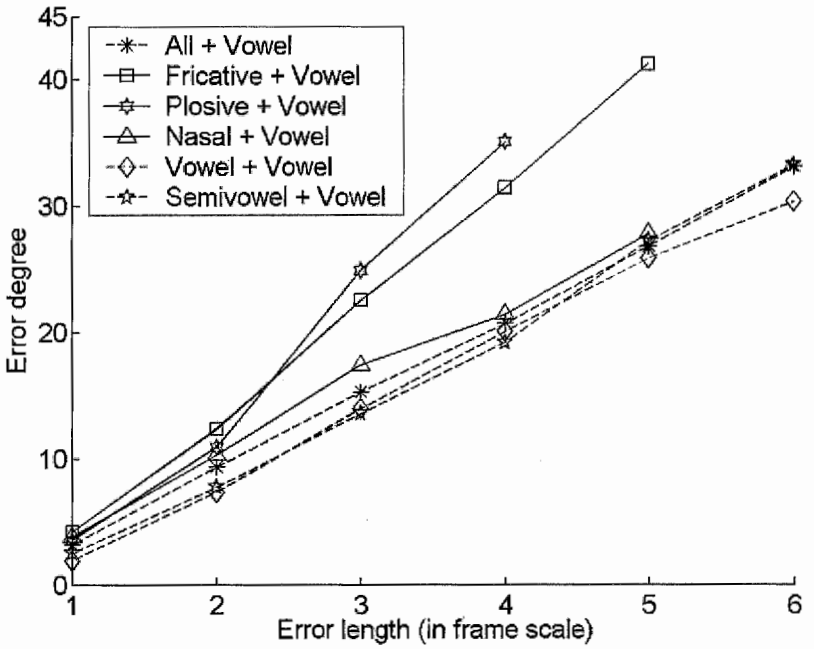


Figure I-1. Correlation between error degree and error length

The HMMs trained by MLE criteria were used to segment the Japanese training data (the details of the data information can be found in section 4). The correlation between error degree and error length was analyzed from all segmentation errors, and the correlations of some typical boundaries are shown in Figure I-1. From the figure, error degree is nearly linear with error length, and for different boundary types, the slope is different, i.e. the correlation is context dependent. For the boundary between plosive and vowel, or fricative and vowel, the slope is relative large, i.e. error degree is sensitive to error length. For the boundary between vowel and vowel, or semivowel and vowel, the slope is relative small, i.e. error degree is less sensitive to error length. This characteristic is identical to the requirement of concatenative speech synthesis, which is quite sensitive to the segmentation accuracy of plosive segments, since a plosive segment with an imprecise boundary might result in two bursts or no burst in synthetic speech, and less sensitive to the accuracy of vowel segment. In this sense, error degree is a meaningful factor for measuring the segmentation error. Because of the correlation between error degree and error length, minimization of error degree is also related to minimizing error length.

1.4 Loss function definition

To consider both explicit error length and inherent error degree, we defined the loss function as

$$\ell(\Lambda) = E_\ell^\alpha E_d = E_\ell^\alpha (g_b(X,\Lambda) - g_c(X,\Lambda)), \tag{I-9}$$

where E_ℓ is error length and α is a positive number. In this loss function, E_ℓ^α is regarded as a constant number in the optimization procedure by the GPD algorithm, so the loss function can be differentiated with respect to the parameters. The meaning of E_ℓ^α can be explained as follows.

On the one hand, it indicates the consideration of explicit error length. When α is larger than 0, the loss of the training data with large error length is large, and accordingly the model parameters are updated on a large scale, which means there is more focus on eliminating large errors. From this point of view, the loss function provides a flexible way to optimize the parameter for the different focus. On the other hand, E_ℓ^α means the weight of the training data, i.e. the same performance can be achieved by repeating the training data E_ℓ^α times when the loss function is defined as E_d only.

This definition of loss function is much more meaningful, reflecting both the explicit error length and the inherent error degree. Moreover, by this definition, the loss function is continuous, differentiable, and directly related to the parameters of HMM. By using the gradient-based optimization method (e.g. GPD), the loss function can be minimized, which relates to a minimization of the segmentation error.

1.5 Parameter updating

Next, we optimized the parameters under this loss function by the GPD algorithm. For a state j of HMM h which has M mixtures, the output probability distribution is

$$b_{h,j}(x_t) = \sum_{m=1}^{M} c_{h,j,m} b_{h,j,m}(x_t) = \sum_{m=1}^{M} c_{h,j,m} G\left[x_t; \mu_{h,j,m}, R_{h,j,m}\right], \tag{I-10}$$

where $b_{h,j,m}(\cdot)$ is the output probability of one mixture, $G[\cdot]$ is a normal Gaussion distribution, and $c_{h,j,m}$, $\mu_{h,j,m} = [\mu_{h,j,m,l}]_{l=1}^{D}$ and $R_{h,j,m} = [\sigma_{h,j,m,l}]_{l=1}^{D}$ are mixture weights, mean vector and covariance matrix, respectively.

It should be noted that the HMM as a probability measure has some original constraints, such as: 1) the function is positive; 2) $\sum_m c_{h,j,m} = 1$ for all h, j, and 3) $\sigma_{h,j,m,l} > 0$. In order to maintain these constraints during parameter adaptation, we should take some parameter transformations as follows:

$$c_{h,j,m} \to \tilde{c}_{h,j,m} \quad \text{where} \quad c_{h,j,m} = \frac{\exp(\tilde{c}_{h,j,m})}{\sum_k \exp(\tilde{c}_{h,j,m})} \tag{I-11}$$

$$\mu_{h,j,m} \to \tilde{\mu}_{h,j,m} = \mu_{h,j,m} R_{h,j,m}^{-1} \tag{I-12}$$

$$R_{h,j,m} \to \tilde{R}_{h,j,m} = \log(R_{h,j,m}) \tag{I-13}$$

The transformation in (12) is important for designing the step size for convergence. More discussion about the parameter transformation can be found in [6].

For a sample X_n in the training set, the adaptation of the parameter is

$$\Lambda_{h,j,m}(n+1) = \Lambda_{h,j,m}(n) - \varepsilon \frac{\partial \ell(X_n;\Lambda)}{\partial \Lambda_{h,j,m}}\Big|_{\Lambda=\Lambda_n}, \tag{I-14}$$

where

$$\frac{\partial \ell(X;\Lambda)}{\partial \Lambda_{h,j,m}} = E_\ell^\alpha \frac{\partial (g_b(X,\Lambda) - g_c(X,\Lambda))}{\partial \Lambda_{h,j,m}}$$

$$= E_\ell^\alpha \sum_{t=1}^{T} (\delta(q_{bt} - j) - \delta(q_{ct} - j)) b_{h,j}^{-1}(x_t) \frac{\partial b_{h,j}(x_t)}{\partial \Lambda_{h,j,m}}, \tag{I-15}$$

where $\delta(\cdot)$ denotes the Kronecher delta function. For mean vector, the updating rule is

$$\frac{\partial b_{h,j}(x_t)}{\partial \tilde{\mu}_{h,j,m}} = c_{h,j,m} b_{h,j,m}(x_t) R_{h,j,m}^{-1}(x_t - \mu_{h,j,m}). \tag{I-16}$$

Finally,

$$\mu_{h,j,m}(n+1) = \tilde{\mu}_{h,j,m}(n+1) R_{h,j,m}. \tag{I-17}$$

Similarly, for the covariance matrix $R_{h,j,m}$, the updating rule is

$$\frac{\partial b_{h,j}(x_t)}{\partial \tilde{R}_{h,j,m}} = c_{h,j,m} b_{h,j,m}(x_t) \cdot \left(R_{h,j,m}^{-1} R_{h,j,m}^{-1} (x_t - \mu_{h,j,m})(x_t - \mu_{h,j,m})^T - I_D \right), \tag{I-18}$$

where I_m is a identity matrix. Finally,

$$R_{h,j,m}(n+1) = \exp\{\tilde{R}_{h,j,m}(n+1)\}. \tag{I-19}$$

Also, the mixture weight is updated as

$$\frac{\partial b_{h,j}(x_t)}{\partial \tilde{c}_{h,j,m}} = b_{h,j,m}(x_t) c_{h,j,m}(1 - c_{h,j,m}) \tag{I-20}$$

Finally

$$c_{h,j,m}(n+1) = \frac{\exp(\tilde{c}_{h,j,m}(n+1))}{\sum_k \exp(\tilde{c}_{h,j,m}(n+1))}. \tag{I-21}$$

The meaning of the updating rule can be explained as follows. In equation (15), $\delta(q_{bt} - j) - \delta(q_{ct} - j)$ is equal to zero when $q_{bt} = q_{ct}$, or equal to 1 when $q_{bt} \neq q_{ct}$ and $q_{bt} = j$, or equal to –1 when $q_{bt} \neq q_{ct}$ and $q_{ct} = j$, which indicates, for an input vector, if the best state alignment differs with the correct state alignment, the updating rule is to move the parameters of the incorrect state model far away from the vector and to move the parameters of the correct state model close to the vector.

2 Explicit duration modeling for automatic segmentation

2.1 Motivation

After applying the discriminative training method, the performance of HMM-based segmentation was quite impressive with small average of total segmentation errors, but it is not perfect due to the inevitability of large segmentation errors, which would largely decrease the quality of synthetic speech.

Commonly, the segmentation error is defined as the location difference of the boundary between humanly labeled and automatically labeled. By examining the segmentation result, we found that the phonemes with large segmentation errors often have implausible duration, which is rather longer or smaller than the natural duration. This phenomenon promotes to consider the adoption of explicit duration model to avoid this kind of errors. In conventional HMM, the duration model does not explicitly exist, only implied by the transition probability, which assume the duration probability distribution of each state as a geometric distribution [9]. In deed, this distribution is usually inappropriate. Instead, several approaches to duration modeling have been proposed to for speech recognition and improve the performance in a certain extent [10][11][12]. Due to the difference between speech recognition task and AS task, we should find appropriate duration modeling for AS.

2.2 Explicit duration modeling

In conventional HMM, the duration model is implied by the transition probability, assuming the duration probability distribution of each state as a geometric distribution, i.e. $p_i(\tau) = a_{ii}^{\tau-1}(1-a_{ii})$, where i is the state, $p_i(\tau)$ is the state duration, and $p_i(\tau)$ is the state self transition probability [9]. In deed, the exponential distribution is usually inappropriate to the actual duration distribution [11]. Due to this, the large error with abnormal duration can occur in HMM-based automatic segmentation. In order to solve this problem, it is necessary to combine explicit duration model with acoustic model.

Let us see the segmentation procedure of an input vector sequence X with N segments. For a state alignment $q = \{q_1, q_2, \cdots, q_N\}$, where each state sequence q_i corresponds to one segment, the vector sequence is accordingly divided to $X = (X_1, X_2, \cdots, X_N)$. Without explicit duration model, the conventional likelihood is calculated by:

$$L_{con}(X, q; \Lambda) = \sum_{i=1}^{N} \log(P_i^a(X_i, q_i; \Lambda_a)) \tag{I-22}$$

where $P_i^a(X_i, q_i; \Lambda_a)$ is the output probability of acoustic model. The transition probability is included in the acoustic model. Combined the acoustic model with the duration model, the new likelihood is calculated by:

$$L_{EDM}(X, q; \Lambda) = \sum_{i=1}^{N} \left(\log(P_i^a(X_i, q_i; \Lambda_a)) + w_i^d \log(P_i^d(T_i; \Lambda_d)) \right) \tag{I-23}$$

where $P_i^d(T_i;\Lambda_d)$ and w_i^d are the output probability of duration model and weight of duration model, and T_i is the length of q_i, i.e. the length of each segment. Under the new definition of likelihood calculation, the segmentation procedure is to find the optimal state alignment, which satisfies:

$$\tilde{q} = \max_{q} L_{EDM}(X,q;\Lambda)$$

$$= \max_{q_i,T_i} \sum_{i=1}^{N} \left(\log(P_i^a(X_i,q_i;\Lambda_a)) + w_1^d \log(P_i^d(T_i;\Lambda_d))\right), \quad \text{(I-24)}$$

where $\tilde{q}$ is the best state sequence.

For duration model, one important thing is to choose the carrier of duration model, which might be state, phone, or syllable etc. In order to directly relate to the aim of segmentation, the duration model is based on the basic segment, i.e. if the basic segment is phone, the duration model is phone duration model, etc. In segmentation, the duration can be regarded as either discrete variance in frame scale or continuous variance in time scale. Accordingly, the probability distribution of duration model can be Gammar or Gaussian distribution [11]. Here, we adopt the Gaussion distribution, i.e.

$$P^d(T;\Lambda_d) = \frac{1}{\sqrt{2\pi v^d}} \exp\left(-\frac{1}{2}\left(\frac{T-m^d}{v^d}\right)^2\right) \quad \text{(I-25)}$$

where m^d and v^d are respectively the mean and variance of duration model.

2.3 Two-step based segmentation method

One reason that explicit duration model was not adopted in conventional HMM is the problem of computational complexity. If we directly search the optimal state alignment based on equation (3), the computational cost of segmentation is excessive large. To reduce the computational cost to endurable degree, many methods have been proposed for speech recognition [12][13], where the temporal constraint and path pruning were used. As these methods are not designed for segmentation, here, we proposed a two-step-based method to perform the segmentation with duration model, where the duration model is incorporated in a postprocessor procedure. The detailed procedure is described as follows.

1). First step

In first step, the explicit duration model is not taken account into segmentation. Without duration model, the DP algorithm can be performed with high efficiency, which is similar to conventional method.

2). Second step

Based on the segmentation result of first step, combining duration model, we use heuristic technique to search the optimal path by iterative procedure. It performed as:

a) From the first boundary to the last of the sentence, search the optimal position by following operation.
b) Shift the boundary to the left and calculate the new likelihood combined with duration model. If the likelihood increased, perform b) again; else go to c).
c) Shift the boundary to the right and calculate the new likelihood combined with

duration model. If the likelihood increased, perform c) again; else go to d).

d) If there is no allowable shift performed in whole sentence, the procedure finished. Otherwise, go back to a).

Please notice that the new likelihood is calculated only with the segment inside the window centered at the boundary, which means the boundary is optimized locally in each step. After several iterations of the local optimization, the result will be convergent and close to the global optimum. The preliminary experiments showed that the optimization result with different window sizes have a little difference. However, it also showed that the difference could be compensated by the appropriate weights. For high efficiency, the window size adopted in the later experiments is fixed to 2.

Compared to the conventional computational cost of segmentation, the extra cost of this method is the cost of second step. In the second step, the computational cost of one loop is comparable to the conventional cost. Commonly, it take about 2 ~ 5 iterations to converge to the optimum result. Then, the total computational cost of this method is just several times of conversional cost, which is endurable for segmentation as it is an off-line task.

2.4 Weight optimization

From the result of former experiment, the effect of duration model critically depends on aptness of the weight coefficient. And for different phonemes, the effect of duration model is different. Hence, it is very important to optimize the weight coefficient for each phoneme. As the duration model is incorporated as a postprocessor and the segmentation is based on local optimization, the conventional gradient-based optimization methods are not suitable for this task. Here, we adopted the hill-climbing algorithm to optimize the weight coefficients of duration model. The detail is as follows.

a) Decrease or increase the weight coefficient of one phoneme in a certain strategy until the segmentation accuracy has no more improvement.

b) Perform a) for each phoneme.

c) If there is no allowable modification on the weight coefficients for all phonemes, then stop. Otherwise, perform b) again.

As we expected, the weights of duration model for each phoneme are different after optimization.

3 Experiments

3.1 Experimental condition

Here, we examine the effect of discriminative training and the explicit duration modeling. The experiments were performed both on Chinese and Japanese data. The details information of training and testing are as follows:

1) *Chinese:* The training and testing data consists of 1000 and 680 sentences, including 27312 and 15872 phones respectively, and all the data had been hand-labeled. The phone set has 60 phonemes, including 21 initials, 37 finals,

pause and silence. Monophone HMMs are adopted and the number of state is three for initials, pause and silence, and five for finals, and the number of mixture set to five for each phonemes. The acoustic feature is 16 orders MFCC and energy, and the delta coefficients. The analysis window size and shift are 20ms and 5ms respectively.

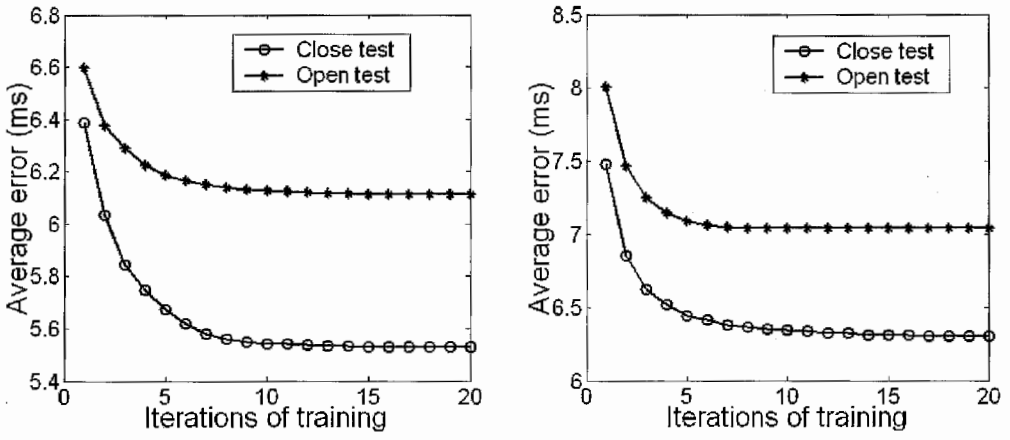
2) *Japanese*: The training and testing data consists of 2263 and 501 phonetically balanced sentences, including 185404 and 30706 phones respectively. The phone set used here includes 60 phonemes. Also monophone HMMs are used, and the number of state and mixture are respectively three and five for each phoneme. The configuration of acoustic feature analysis is the same to that on Chinese data.

3.2 Effect of MSGE-based discriminative training

We trained the HMMs by using MLE and MSGE criterion and then compared the segmentation accuracies of these two methods. The MLE-based HMM training is performed by the HTK tools.[8] In MSGE-based training, the HMMs are initialized by the results of MLE-based training. The performance was evaluated on both Chinese and Japanese data.

3.2.1 Effect on Chinese data

From the result of close and open test in Figure I-2(a), the MSGE-based discriminative training is convergent after 10-20 iterations. As can be seen in Table I-1, the accuracy of segmentation improved after MSGE-based training, especially for the errors less than 5ms. We also examined the effect of error length on loss function by training with different α values. When α increases from 0 to 1, which means we have more focus on larger errors, the percentage of the errors less than 30 ms increased 0.13 %, whereas the percentage of error less than 5ms decreased 0.83%.



(a)Chinese data (b) Japanese data

Figure I-2. Convergence of MSGE-based discriminative training

The details on accuracy with different phonetic boundaries are shown in Table I-2. After MSGE-based training, the average error of the CV-boundary decreased from 4.51 ms to 3.60 ms, i.e. a reduction of 19.7%, whereas that of the VV-boundary decreased 9.6%.

Since we noted that concatenative speech synthesis is much more sensitive to the accuracy of the CV-boundary and insensitive to the VV-boundary, this improvement appears to be reasonable for speech synthesis.

Table I-1. Segmentation accuracy for Chinese

	Percentage of the accuracy (%)				Aver
	≤5ms	≤10ms	≤20ms	≤30ms	(ms)
MLE	70.44	86.89	95.58	97.75	6.856
MSGE(α=0)	76.01	88.70	95.74	97.85	6.112
MSGE(α=1)	75.18	88.65	95.81	97.98	6.174

Table I-2. Accuracy with different phonetic boundaries (Chinese)

	Average error (ms)			
	CC	CV	VC	VV
MLE	×	4.51	5.69	11.99
MSGE(α=0)	×	3.60	5.37	10.83

Table I-3. Segmentation accuracy for Japanese

	Percentage of the accuracy (%)				Aver
	≤5ms	≤10ms	≤20ms	≤30ms	(ms)
MLE	60.84	79.64	92.07	96.31	8.666
MSGE(α=0)	70.15	84.46	94.00	97.29	7.035
MSGE(α=1)	69.68	84.40	94.24	97.43	7.084

Table I-4. Accuracy with different phonetic boundaries (Japanese)

	Average error (ms)			
	CC	CV	VC	VV
MLE	5.18	7.85	7.16	11.31
MSGE(α=0)	4.64	4.84	6.45	9.59

3.2.2 Effect on Japanese data

The convergence of MSGE-based discriminative training on Japanese data can be found in Figure I-2(b). From Table I-3, the segmentation accuracy for Japanese was improved after MSGE-based training, and the effect of E_ℓ with different α values is similar to that for Chinese. In Table I-4, the largest improvement also occurred in the accuracy of the CV-boundary, where the average error reduced from 7.85 ms to 4.84 ms, i.e. a reduction of 38%.

Comparing the results for Japanese and Chinese data, we found that the improvement for Japanese is much larger than that for Chinese. One reason is that the HMM modeling in Japanese is not optimized, that is, it simply uses 3-state model for all phonemes. Therefore, the segmentation accuracy of the baseline trained by MLE criteria for Japanese is much worse than that for Chinese. Nevertheless, the difference in accuracy between Japanese and Chinese data is reduced after MSGE-training. This indicates that the MSGE-based training method can work well even when the HMM modeling is not

optimized. Furthermore, it can compensate for the inaccuracy of the HMM modeling to a certain extent.

3.3 Effect of explicit duration modeling

3.3.1 Preliminary experiments

Here, the effect of explicit duration model on segmentation was investigated by applying different weight coefficients. For simplification, the weights of all phonemes are set to the same coefficients. The mean and variance of duration model for each phoneme are initialized with the statistical parameters calculated from the training data. Also, we investigated the effect both on Chinese and Japanese data to examine the language dependency of duration model. It should be noted that the model without duration model, i.e. with acoustic model only, is regarded as the baseline model, and the acoustic HMMs are trained by HTK tools [8].

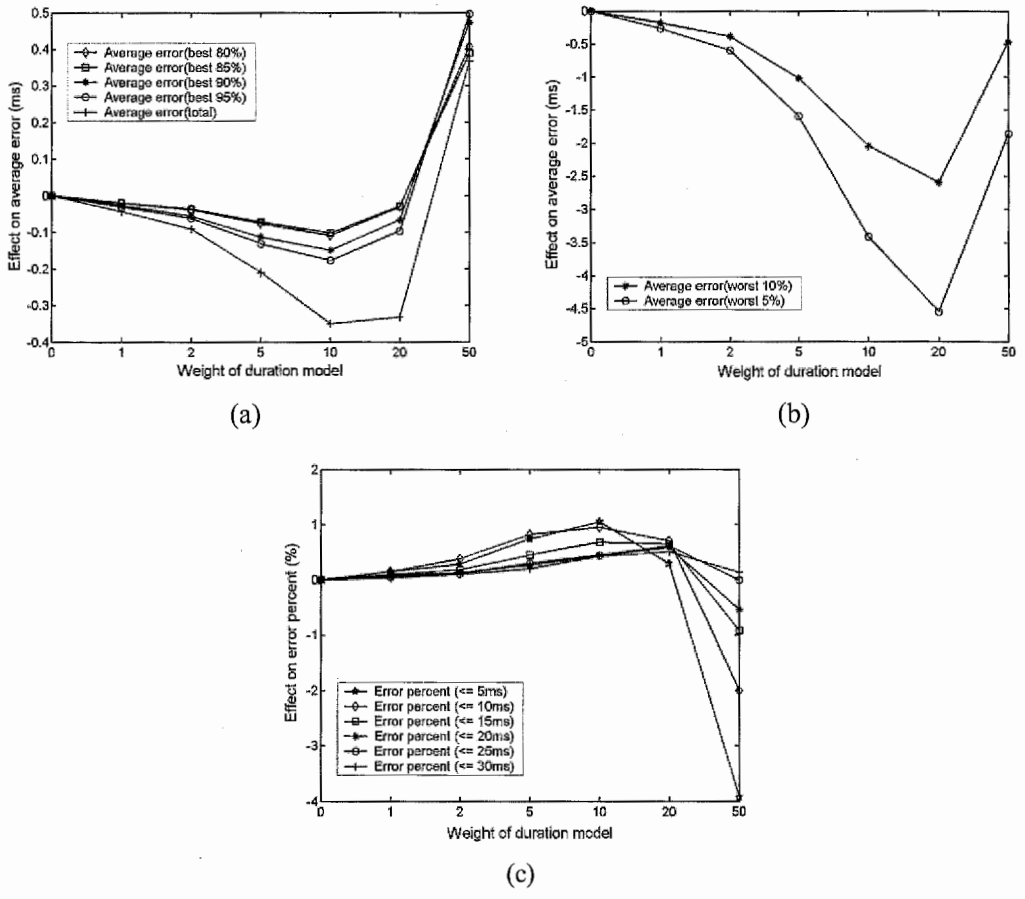


Figure I-3: segmentation accuracy on Chinese data
(a) The error average of the best (*) percent of boundary;
(b) The error average of the worst (*) percent of boundary
(c) The percentage of boundary whose error length less than (*) ms

3.3.1.1 Effect on Chinese data

The effect of duration model on Chinese data is obviously shown with different aspects in Figure I-3. Please note that the value shown in the figure is the difference of the

segmentation accuracy between the current model and the baseline model. After combined with the duration model, the average of the best 80%~100% segmentation errors decreased when the weight increase from 0 to 10, and increased when the weight increase from 10 to 50 in Figure I-3(a). The average of the worst 5% and 10% segmentation errors are shown in Figure I-3(b). From this figure, the tendency of the improvement is similar to Figure I-3(a), except that the average error still decreased when the weight increased from 10 to 20. And the other difference is that the improvement in Figure I-3(b) is much larger than Figure I-3(a). The same phenomena can also be found in Figure I-3(c), where the percentage of the segmentation error less than 5, 10 or 15ms decrease when the weight increases from 10 to 20, whereas the percentage of the segmentation error less than 20, 25 or 30ms still increase. As we expected, these showed the duration model has more effect on improving the large segmentation errors.

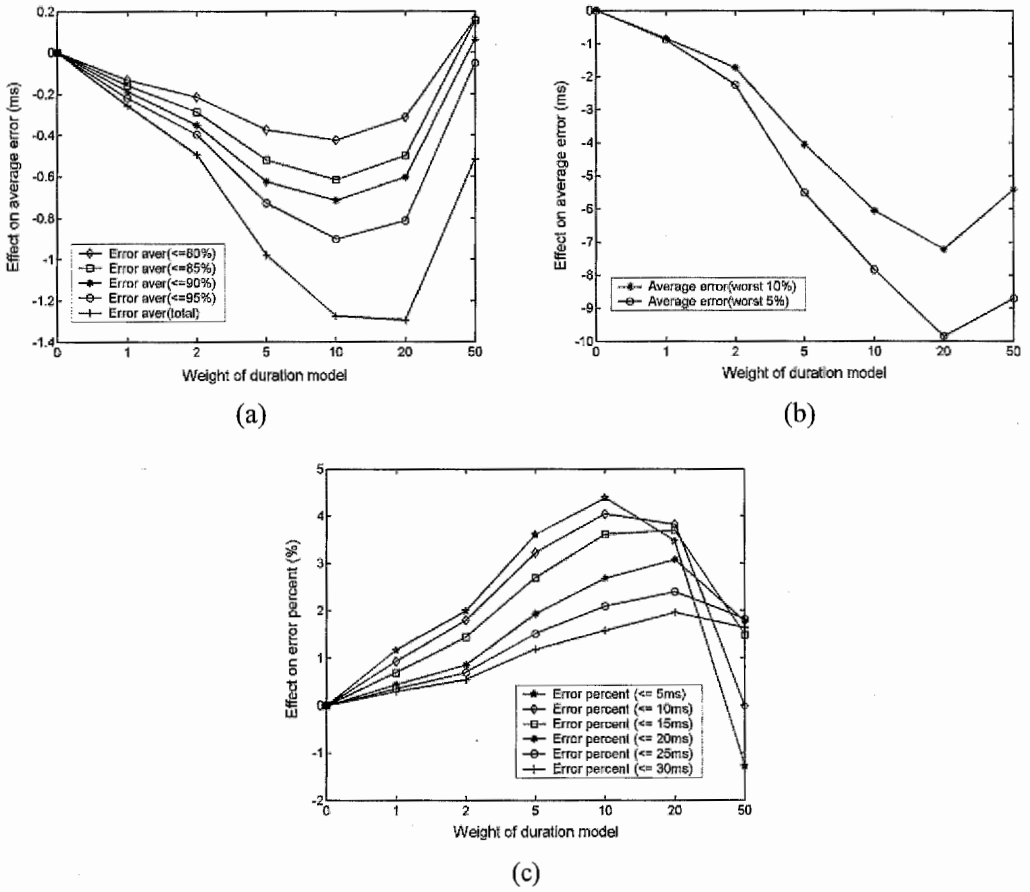


Figure I-4: segmentation accuracy on Japanese data
(a) The error average of the best (*) percent of boundary;
(b) The error average of the worst (*) percent of boundary
(c) The percentage of boundary whose error length less than (*) ms

3.3.1.2 Effect on Japanese data

The effect of duration model on Japanese data is shown in Figure I-4, which is similar to that on Chinese data. Also the duration model has more effect on improving the large errors, especially the average of the worst 5% segmentation errors reduced nearly 10ms when the weight is 20. The only difference is that the effect of duration model on Japanese

is much larger than that on Chinese. The reason is that the phoneme duration in Japanese is more stable than Chinese.

From the result both on Chinese and Japanese data, we can find the accuracy of segmentation was improved after combined with the explicit duration model with appropriate weights. And the effect of duration model is language dependant. If the phoneme duration in the language is much stable, the duration model has much effect, and vise verse. Nevertheless, the duration model has much effect on improving the segmentation errors, especially eliminating the large errors.

3.3.2 Results after weights optimization

Finally, we performed the weight optimization both on Chinese and Japanese model and the results are shown in Table I-5 and Table I-6 respectively. From the results, the error average of worst 5% was reduced 4.95ms and the number of the errors larger than 30ms was reduced 27% on Chinese. Also, the error average of worst 5% was reduced 9.88ms and the number of the errors larger than 30ms was reduced 47% on Japanese. These indicate the segmentation accuracy was improved after combining the optimized duration model, especially the large errors was largely eliminated.

Table I-5. The segmentation result on Chinese data

	Error aver (ms)		Error percent (%)	
	Best 95%	Worst 5%	>20ms	>30ms
Baseline	5.26	36.58	4.42	2.25
EDM	4.85	31.63	3.58	1.63

Table I-6. The segmentation result on Japanese data

	Error aver (ms)		Error percent (%)	
	Best 95%	Worst 5%	>20ms	>30ms
Baseline	6.86	42.31	7.92	3.69
EDM	5.85	32.43	4.74	1.75

Table I-7. Final segmentation accuracy (MSGE & EDM)

	Error aver (ms)		Error percent (%)		
	Total	Worst 5%	>10ms	>20ms	>30ms
Chinese	5.79	31.78	10.72	3.64	1.68
Japanese	6.61	32.48	14.75	4.70	1.73

Table I-8. Final accuracy on different phonetic boundary

	Error average (ms)			
	CC	CV	VC	VV
Chinese	×	3.63	4.99	9.89
Japanese	4.34	4.47	6.12	8.57

3.4 Combination of two techniques

From the evaluation results of MSGE-based training and explicit duration modeling, these two techniques can improve the segmentation accuracy with different focus, where the MSGE-based training focus on improve the accuracy of the sensitive boundary, e.g. the boundary between the plosive and vowel, and the explicit duration modeling focus on eliminating the large errors. Finally, we applied both two techniques to improve the HMM-based segmentation accuracy. The results are shown in Table I-7 and Table I-8.

By comparing the results with that of previous experiments, we can find that the final results have both advantage of MSGE-based training and explicit duration modeling, i.e. the segmentation accuracy of sensitive boundary had much improvement and the number of large errors was largely decreased. The final average errors of segmentation on Chinese and Japanese are 5.79ms and 6.61ms respectively, which is much better than that of the baseline.

4 Summary

From the results, MSGE-based training and explicit duration modeling can improve the segmentation accuracy with different focus, where the MSGE-based training focus on improve the accuracy of the sensitive boundary, and the explicit duration modeling focus on eliminating the large errors. After combining these two techniques, it has both advantages and the segmentation accuracy was largely improved. However, the segmentation accuracy is not perfect compared to the human labeling. Even though the explicit duration model was applied, there are still some inevitable large errors.

II. HMM-based Chinese prosody modeling

1 Background

The HMM is a widely used statistic model, and has been successful on speech recognition. In recent years, the HMMs have been applied to speech synthesis, and several HMM-based speech synthesis methods had been proposed [14][15][16]. In our method, the synthetic speech is generated from HMM themselves by using a speech parameter generation algorithm[18], which is performed with the dynamic feature constraints. One of the advantage of this HMM-based method is that the voice characteristic of synthetic speech can be changed by transforming the HMM parameters appropriately. Also a simultaneous modeling for the spectrum, pitch and duration was introduced [17], where the feature is composed of the spectrum and the pitch, and the HMM based on multi-space probability distribution (MSD-HMM) was proposed for pitch pattern modeling [21][22].

This HMM-based method had been applied for Japanese prosody modeling, and the results were quite impressive. Here we want to apply this method for Chinese prosody modeling.

2 Techniques

2.1 MSD-HMM based pitch modeling

Here is a brief introduction of MSD-HMM. In the MSD-HMM[21], a sample space Ω is considered, which consists of G spaces:

$$\Omega = \sum_{g=1}^{G} \Omega_g , \tag{II-1}$$

where Ω_g is an n_g dimensional real space R^{n_g}, specified by space index g. For each space Ω_g, it has a probability w_g, i.e. space weight. If $n_g > 0$, each space has a pdf function $P_g(x)$, $x \in R^{n_g}$. We assume that Ω_g contains only one sample point if $n_g = 0$.

The observation feature is represented by a random vector o, which consists of a set of space indexes X and a continuous random variable $x \in R^n$, i.e. $o = (X, x)$. For a N-state MSD-HMM, the state output probably distribution is defined as $B = \{b_i(\cdot)\}_{i=1}^{N}$, where

$$b_i(o) = \sum_{g \in S(o)} w_{ig} P_{ig}(V(o)), \tag{II-2}$$

and

$$S(o) = X, \quad V(o) = x. \tag{II-3}$$

It is noted that we define $P_g(x) \equiv 1$ for $n_g = 0$.

As the observation of F0 has a continuous value in the voiced region, where exist no values for the unvoiced region. We applied the MSD-HMM for F0 modeling, assuming that the observed F0 value occurs from one-dimensional spaces and the "unvoiced" symbol occurs from the zero-dimensional space. By setting $n_g = 0$ ($g = 1,2,...,G-1$), $n_G = 0$ and

$$S(o_t) = \begin{cases} \{1,2,...,G-1\}, & (voiced) \\ \{G\}, & (unvoiced) \end{cases} \tag{II-4}$$

the MSD-HMM can cope with F0 patterns including the unvoiced region without heuristic assumptions. In this case, the observed F0 value is assumed to be drawn from a continuous $(G-1)$-mixture pdf. More details about the MSD-HMM for pitch modeling can be found in [21][22].

2.2 HMM training procedure

The HMM training is performed using the HTS toolkit, which is the modified version of HTK toolkit. The procedure is shown in Figure II-1.

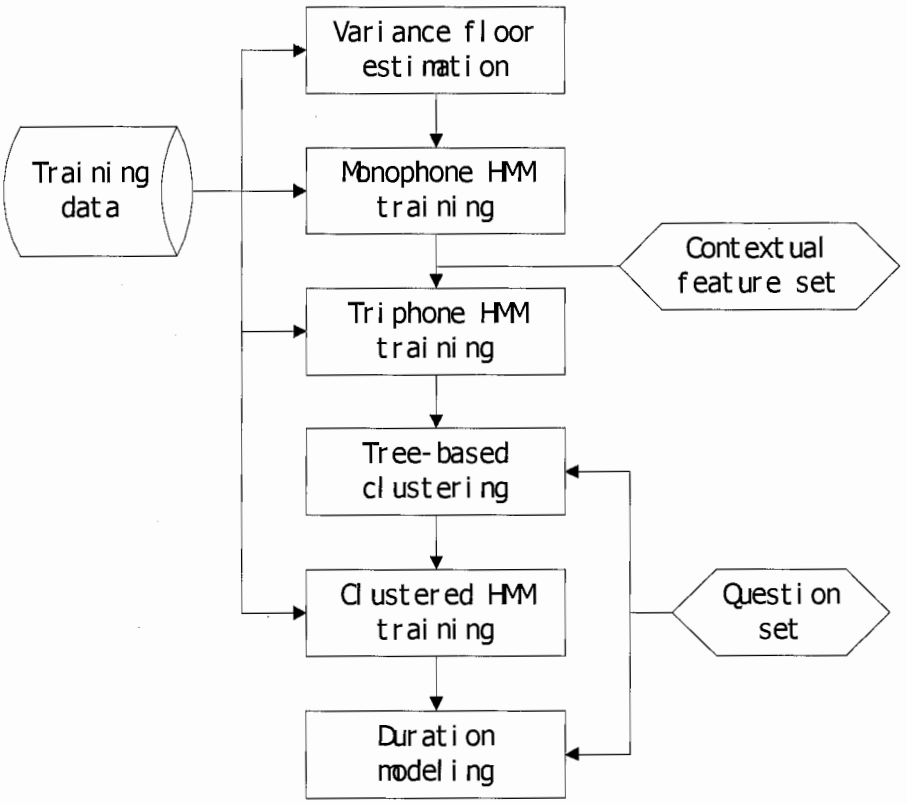


Figure II-1. HMM training procedure

a. *Variance floor estimation*: In order to prevent the variance parameter of HMMs from near to zero when there is few training data, which is common for the full context HMM training, we should set corresponding floor values for them. As the spectrum and f0 parameters with the delta coefficients are used to construct the

MSD-HMM, we should estimate different variance floor for different parameters. Here, the HCompV tool is used to perform this task.

b. *Monophone HMM training*: This step is performed in the common way by using HRest tool.
c. *Full context HMM training*: This step is performed in the common way by using HERest tool. As we adopted many contextual features, which result in a larger number of the full context models, it should be noted that there is only few training sample for each model.
d. *Tree-based clustering*: As there is only one or two training samples for each full context model, the HMM parameter is overfit to the training data. Then the tree-based clustering procedure is very important to improve the robustness of the HMMs and balance the model complexity and the training data. And my work focused on this process.
e. *Clustered HMM training*: This step is performed using HERest tool. At the same time, the statistic information about the occupancy count of each state is output for duration modeling.
f. *Duration modeling*: With the statistic information of the occupancy count, we construct the duration model using the tree-based clustering technique.

2.3 Parameter generation algorithm

With the trained HMM, the next work is to to generate the parameters. Let $O=\{o_1,o_2,...,o_T\}$ be the vector sequence and $q=\{q_1,q_2,...,q_T\}$ be the state sequence. Here, the vector o_t consists of the static feature vector c_t and the dynamic feature vector Δc_t, that is $o_t=\{c_t,\Delta c_t\}$. Then the problem is to dermine the parameter sequence $c=\{c_1,c_2,...,c_T\}$, which maximizes

$$P[O|\lambda]=\sum_{all\ q}P[q,O|\lambda] \tag{II-5}$$

To solve this problem, first consider maximizing $P[q,O|\lambda]$ for a given state sequence q with respect to c. Then the probability $P[q,O|\lambda]$ can be written as

$$P[q,O|\lambda]=P[q|\lambda]\cdot P[O|q,\lambda] \tag{II-6}$$

where

$$P[O|q,\lambda]=b_{q_q}(o_1)b_{q_2}(o_2)\cdots b_{q_T}(o_T) \tag{II-7}$$

and

$$b_j(o_t)=N(c_t;\mu_j,U_j)\cdot N(\Delta c_t;\Delta\mu_j,\Delta U_j) \tag{II-8}$$

where $N(\cdot)$ is the Gaussian distribution, μ_j and U_j are the mean and variance of the static features, and μ_j and U_j are the mean and variance of the dynamic features.

Then an approximation solution for this problem is given in [14]. By using the dynamic features, it performed in an iterative way with high efficiency. The details of this solution can be found in [14].

3 Application for Chinese prosody modeling

Here, we applied the HMM-based method for the Chinese prosody modeling. Based on the Chinese characteristic, we designed corresponding contextual features and question set. Also we improved the tree-based clustering procedure by considering the space weight and the meaning of questions.

3.1 Contextual features

In Chinese speech synthesis, the basic unit is the half-syllable, i.e. initial and final. The prosodic levels are classified as half-syllable, syllable, word, phrase and sentence level. And we design the contextual features and questions for each prosodic level. Here is brief introduction of the contextual features. The details can be found in [30](ATR Technique Report, TR-SLT-0032, Heiga Zen etc).

3.1.1 Initial and final

The classification of the initial and final are shown in Table II-1 and Table II-2.

Table II-1. Classification of initials

Stops	b, d, g
Aspirated stops	p, t, k
Affricatives	z, zh, j
Aspirated affricatives	c, ch, q
Nasals	m, n
Fricatives	f, s, sh, r, x,h
Laterals	l

Table II-2. Classification of finals

Mono finals	a, o, e, i, u, v, ii, iii, er
Compound finals	ai, ei, ao, ou, ia, ie, ua, uo, ve, iao, iou, uai, uei
Nasal finals	an, en, in, vn, ang, eng, ing, ong, ian, uan, van, uen, iang, uang, ueng, iong

Based on the classification, we have three features:

a. Current half-syllable type
b. Previous half-syllable type
c. Next half-syllable type

3.1.2 Syllable

The tone type related to syllable has an important role in Chinese spoken language. The tone has five types: high, low, rise, false and light. And it should be noted that one tone

can change to another tone type, i.e. new tone, in a certain environment. Thus we have two features related to tone type: dictionary tone type and new tone type. The contextual features related to syllable are designed as follows:

a. Previous tone type
b. Previous new tone type
c. Current tone type
d. Current new tone type
e. Next tone type
f. Next new tone type
g. Position of current syllable in the word

3.1.3 Word

In Chinese, the word is the basic carrier for the part-of-speech (POS). The classification of POS is shown in Table II-3. (Different from the classification in [30])

Table II-3. Classification of POS for Chinese

Content	Real	Numeral	Neqa,Neqb,Neu
		Measure	Nf
		Pronoun	Nh,Nep
		Noun	Na,Nl,Ni,Nd,Nc,Nb,Ncd
	Virtual	Adjective	A,Nes
		Adverb	D,Da,Dfa,Dfb,Di,Dk
		Verb	SHI,VA,VAC,VB,VC,VCL,VC,VE, VF,VG,VH,VHC,VI,VJ,VK,VL,V_2
Function		Conjunction	Caa,Cab,Cba,Cbb, P
		Auxiliary	DE,Ng,I,T
Other		Special symbol	BOS,EOS,FW

Based on the classification, the features related to word level are as follows:

a. POS type of previous word
b. POS type of current word
c. POS type of next word
d. Syllable number in previous word
e. Syllable number in current word
f. Syllable number in next word
g. Position of current word in the phrase by {syllable, word} interval

3.1.4 Phrase

The contextual features related to phrase level are as follows:

a. {Syllable, Word} number in previous phrase
b. {Syllable, Word} number in current phrase
c. {Syllable, Word} number in next phrase

d. {Syllable, Word} number in previous phrase
e. Position of current phrase in the sentence by {syllable, word, phrase} interval

3.1.5 Sentence

The contextual features related to sentence level are:

a. {Syllable, Word, Phrase} number in the sentence
b. Sentence type: question, statement

3.2 Question set

Based on these contextual features, the question set used for tree-based clustering was designed. The full list of the questions is shown below. It is similar to the question list in [30](ATR Technique Report, TR-SLT-0032, Heiga Zen etc).

3.2.1 Questions related to half-syllable

{Previous, Current, Next} half-syllable is voiced?
{Previous, Current, Next} half-syllable is final?
{Previous, Current, Next} half-syllable is mono-final?
{Previous, Current, Next} half-syllable is bi-final?
{Previous, Current, Next} half-syllable is tri-final?
{Previous, Current, Next} half-syllable is compound final?
{Previous, Current, Next} half-syllable is nasal final?
{Previous, Current, Next} half-syllable is front final?
{Previous, Current, Next} half-syllable is back final?
{Previous, Current, Next} half-syllable is middle final?
{Previous, Current, Next} half-syllable is open final?
{Previous, Current, Next} half-syllable is i-class final?
{Previous, Current, Next} half-syllable is close final?
{Previous, Current, Next} half-syllable is front lingual final?
{Previous, Current, Next} half-syllable is back lingual final?
{Previous, Current, Next} half-syllable is middle lingual final?
{Previous, Current, Next} half-syllable is lingual apical final?
{Previous, Current, Next} half-syllable is gutturalize?
{Previous, Current, Next} half-syllable is fricative?
{Previous, Current, Next} half-syllable is plosive?
{Previous, Current, Next} half-syllable is affricative?
{Previous, Current, Next} half-syllable is nasal?
{Previous, Current, Next} half-syllable is liquid?
{Previous, Current, Next} half-syllable is labial?
{Previous, Current, Next} half-syllable is labiodental?
{Previous, Current, Next} half-syllable is dental?
{Previous, Current, Next} half-syllable is apical inital
{Previous, Current, Next} half-syllable is apical front?

{Previous, Current, Next} half-syllable is apical back?
{Previous, Current, Next} half-syllable is velar?
{Previous, Current, Next} half-syllable is unvoiced?
{Previous, Current, Next} half-syllable is voiced fricative?
{Previous, Current, Next} half-syllable is unvoiced fricative?
{Previous, Current, Next} half-syllable is voiced plosive?
{Previous, Current, Next} half-syllable is unvoiced plosive?

{Previous, Current, Next} half-syllable is /a/?
{Previous, Current, Next} half-syllable is /an/?
{Previous, Current, Next} half-syllable is /ang/?
{Previous, Current, Next} half-syllable is /ao/?
{Previous, Current, Next} half-syllable is /b/?
{Previous, Current, Next} half-syllable is /c/?
{Previous, Current, Next} half-syllable is /ch/?
{Previous, Current, Next} half-syllable is /d/?
{Previous, Current, Next} half-syllable is /e/?
{Previous, Current, Next} half-syllable is /ei/?
{Previous, Current, Next} half-syllable is /en/?
{Previous, Current, Next} half-syllable is /eng/?
{Previous, Current, Next} half-syllable is /er/?
{Previous, Current, Next} half-syllable is /f/?
{Previous, Current, Next} half-syllable is /g/?
{Previous, Current, Next} half-syllable is /h/?
{Previous, Current, Next} half-syllable is /i/?
{Previous, Current, Next} half-syllable is /ii/?
{Previous, Current, Next} half-syllable is /iii/?
{Previous, Current, Next} half-syllable is /ia/?
{Previous, Current, Next} half-syllable is /ian/?
{Previous, Current, Next} half-syllable is /iang/?
{Previous, Current, Next} half-syllable is /iao/?
{Previous, Current, Next} half-syllable is /ie/?
{Previous, Current, Next} half-syllable is /in/?
{Previous, Current, Next} half-syllable is /ing/?
{Previous, Current, Next} half-syllable is /iong/?
{Previous, Current, Next} half-syllable is /iu/?
{Previous, Current, Next} half-syllable is /j/?
{Previous, Current, Next} half-syllable is /k/?
{Previous, Current, Next} half-syllable is /l/?
{Previous, Current, Next} half-syllable is /m/?
{Previous, Current, Next} half-syllable is /n/?
{Previous, Current, Next} half-syllable is /o/?
{Previous, Current, Next} half-syllable is /ong/?
{Previous, Current, Next} half-syllable is /ou/?

{Previous, Current, Next} half-syllable is /p/?
{Previous, Current, Next} half-syllable is /pau/?
{Previous, Current, Next} half-syllable is /q/?
{Previous, Current, Next} half-syllable is /r/?
{Previous, Current, Next} half-syllable is /s/?
{Previous, Current, Next} half-syllable is /sh/?
{Previous, Current, Next} half-syllable is /sil/?
{Previous, Current, Next} half-syllable is /sp/?
{Previous, Current, Next} half-syllable is /t/?
{Previous, Current, Next} half-syllable is /u/?
{Previous, Current, Next} half-syllable is /ua/?
{Previous, Current, Next} half-syllable is /uai/?
{Previous, Current, Next} half-syllable is /uan/?
{Previous, Current, Next} half-syllable is /uang/?
{Previous, Current, Next} half-syllable is /ui/?
{Previous, Current, Next} half-syllable is /un/?
{Previous, Current, Next} half-syllable is /uo/?
{Previous, Current, Next} half-syllable is /v/?
{Previous, Current, Next} half-syllable is /van/?
{Previous, Current, Next} half-syllable is /ve/?
{Previous, Current, Next} half-syllable is /vn/?
{Previous, Current, Next} half-syllable is /x/?
{Previous, Current, Next} half-syllable is /z/?
{Previous, Current, Next} half-syllable is /zh/?

3.2.2 Questions related to syllable

Tone of {previous, current, next} syllable == N?
Tone of {previous, current, next} syllable <= N?

Tone of {previous, current, next} syllable is high?
Tone of {previous, current, next} syllable is low?

Position of current syllable in word from {head, tail} == N (in syllable interval)?
Position of current syllable in word from {head, tail} <= N (in syllable interval)?

Pause exist between current and {previous, next} syllable?

3.2.3 Questions related to word

{Previous, Current, Next} word is content word?
{Previous, Current, Next} word is function word?
{Previous, Current, Next} word is symbol?
{Previous, Current, Next} word is real word?
{Previous, Current, Next} word is virtual word?

{Previous, Current, Next} word is numeral word?
{Previous, Current, Next} word is measure word?
{Previous, Current, Next} word is noun word?
{Previous, Current, Next} word is adjective word?
{Previous, Current, Next} word is adverb word?
{Previous, Current, Next} word is verb word?
{Previous, Current, Next} word is conjunction word?
{Previous, Current, Next} word is auxiliary word?

POS of {previous, current, next} word is "Neqa"?
POS of {previous, current, next} word is "Neqb"?
POS of {previous, current, next} word is "Neu"?
POS of {previous, current, next} word is "Nf"?
POS of {previous, current, next} word is "Nh"?
POS of {previous, current, next} word is "Nep"?
POS of {previous, current, next} word is "Na"?
POS of {previous, current, next} word is "Nl"?
POS of {previous, current, next} word is "Ni"?
POS of {previous, current, next} word is "Nd"?
POS of {previous, current, next} word is "Nc"?
POS of {previous, current, next} word is "Nb"?
POS of {previous, current, next} word is "Ncd"?
POS of {previous, current, next} word is "A"?
POS of {previous, current, next} word is "Nes"?
POS of {previous, current, next} word is "D"?
POS of {previous, current, next} word is "Da"?
POS of {previous, current, next} word is "Dfa"?
POS of {previous, current, next} word is "Dfb"?
POS of {previous, current, next} word is "Di"?
POS of {previous, current, next} word is "Dk"?
POS of {previous, current, next} word is "SHI"?
POS of {previous, current, next} word is "VA"?
POS of {previous, current, next} word is "VAC"?
POS of {previous, current, next} word is "VB"?
POS of {previous, current, next} word is "VC"?
POS of {previous, current, next} word is "VCL"?
POS of {previous, current, next} word is "VE"?
POS of {previous, current, next} word is "VF"?
POS of {previous, current, next} word is "VG"?
POS of {previous, current, next} word is "VH"?
POS of {previous, current, next} word is "VHC?
POS of {previous, current, next} word is "VI"?
POS of {previous, current, next} word is "VJ"?
POS of {previous, current, next} word is "VK"?

POS of {previous, current, next} word is "VL"?
POS of {previous, current, next} word is "V_2"?
POS of {previous, current, next} word is "Caa"?
POS of {previous, current, next} word is "Cab"?
POS of {previous, current, next} word is "Cba"?
POS of {previous, current, next} word is "Cbb"?
POS of {previous, current, next} word is "P"?
POS of {previous, current, next} word is "DE"?
POS of {previous, current, next} word is "Ng"?
POS of {previous, current, next} word is "I"?
POS of {previous, current, next} word is "T"?
POS of {previous, current, next} word is "BOS"?
POS of {previous, current, next} word is "EOS"?
POS of {previous, current, next} word is "FW"?

Length of {previous, current, next} word == N (in syllable interval)?
Length of {previous, current, next} word <= N (in syllable interval)?

Position of current word in phrase from {head, tail} == N (in syllable interval)?
Position of current word in phrase from {head, tail} <= N (in syllable interval)?
Position of current word in phrase from {head, tail} == N (in word interval)?
Position of current word in phrase from {head, tail} <= N (in word interval)?

3.2.4 Questions related to phrase

Length of {previous, current, next} phrase == N (in syllable interval)?
Length of {previous, current, next} phrase <= N (in syllable interval)?
Length of {previous, current, next} phrase == N (in word interval)?
Length of {previous, current, next} phrase <= N (in word interval)?

Position of current phrase in sentence from {head, tail} == N (in syllable interval)?
Position of current phrase in sentence from {head, tail} <= N (in syllable interval)?
Position of current phrase in sentence from {head, tail} == N (in word interval)?
Position of current phrase in sentence from {head, tail} <= N (in word interval)?

{Previous, Current, Next} phrase is question?

3.2.5 Questions related to sentence

Sentence length in {syllable, word, phrase} interval == N?
Sentence length in {syllable, word, phrase} interval < = N?

Sentence type is question?
Sentence type is statement?

3.3 Role of space weight in MSD-HMM

In the MSD-HMM, the F0 is modeled as two-space variable, where one space represents the observed F0 value and another is the "unvoiced" symbol. It is obvious that the space weight is much important which indicates the voiced degree of the model, where the voiced space weight $W_v = \sum_{i=1}^{G-1} w_i$ represents the voiced probability and the unvoiced space weight $W_u = w_G$ represents the unvoiced probability. As the space weight is not taken into account in the conventional HMM, some techniques used in the training and synthesis, are not suitable for the MSD-HMM. Due to this, we should modify them by considering the space weight.

3.3.1 Considering space weight for tree-based clustering

Although the large number of the contextual HMMs can help to capture the variable in speech data, it results in too many free parameters and low robustness of the models. Due to this, the decision tree-based contextual clustering technique has been applied to improve the robustness[23]. Several criteria, including Maximum Likelihood (ML) and Minimum Description Length (MDL) criterion[24], have been proposed to construct the decision tree. However, these criteria are not suitable for current use because of the difference between the MSD-HMM and the conventional HMM.

In ML or MDL criterion, the splitting score is calculated as the increase of the likelihood. For an attempted question, the node N_0 is split as two child nodes N_1 and N_2, and each node has a related HMM, which is estimated from the data in the node. The splitting score, i.e. the increase of the likelihood, is calculated as:

$$S = \Delta L = L_1 + L_2 - L_0, \tag{II-9}$$

where L_0, L_1 and L_2 are respectively the total likelihood of N_0, N_1 and N_2.
In the MSD-HMM, for a random vector $o = (X, x)$, the likelihood is calculated as:

$$\ell(o) = \log(b(o)) = \log\left(\sum_{g \in S(o)} w_g P_g(V(o)) \right). \tag{II-10}$$

In this equation, the effect of the space weight on likelihood calculation is not clear. Lets see a simple case. When the observed F0 value is modeled as single-mixture pdf., i.e. $G = 2$, we can rewrite the equation (5) as:

$$\ell(o) = \begin{cases} \log W_v + \log P_v(V(o)), & (voiced) \\ \log W_u, & (unvoiced) \end{cases}. \tag{II-11}$$

Accordingly, the total likelihood L_i $(i = 0,1,2)$ is calculated as:

$$L_i = \sum_{o \in N_i} \ell(o) = \sum_{o \in N_{v,i}} \log P_v(V(o)) + LS_i, \tag{II-12}$$

where $N_{v,i}$ is the voiced vectors in the node N_i and

$$LS_i = M_{v,i} \log W_{v,i} + M_{u,i} \log W_{u,i}, \quad \text{(II-13)}$$

where $M_{v,i}$ and $M_{u,i}$ is the number of the voiced vectors and unvoiced vectors in the node N_i. By substituting equation (8) and (9) into (5), we get the splitting score

$$S = \Delta L = C_v + C_u + Q(o), \quad \text{(II-14)}$$

where

$$C_v = M_{v,0} \log W_{v,0} - M_{v,1} \log W_{v,1} - M_{v,2} \log W_{v,2}, \quad \text{(II-15)}$$

and

$$C_u = M_{u,0} \log W_{u,0} - M_{u,1} \log W_{u,1} - M_{u,2} \log W_{u,2}. \quad \text{(II-16)}$$

It can be seen that the variation of the component $C_v + C_u$ is small, i.e. it is trivial for the splitting score calculation. This means the space weight has little effect on the tree-based clustering. However, as we mentioned above, the space weight is much important for voiced or unvoiced decision. Accordingly, it should have more effect on the splitting score calculation.

Here, $W_{v,1}$ and $W_{v,2}$ was denoted as the voiced space weight of the node N_1 and N_2, and $W_{v,1} \geq W_{v,2}$. By incorporating the space weight, the new splitting score is calculated as:

$$S' = \Delta L * C_p(W_{v,2}) * C_d(W_{v,1}, W_{v,2}) \quad \text{(II-17)}$$

where

$$C_p(W) = e^{(W - T_{uv}) * \alpha}, \quad \text{if } W < T_{uv} \quad \text{(II-18)}$$

and

$$C_d(W_1, W_2) = e^{\beta (W_1 - W_2)}, \quad \text{(II-19)}$$

where α and β are the const coefficients, and T_{uv} is the threshold of unvoiced/voiced. In this equation, $C_p(w)$ means the penalty score for the "unvoiced node", which prevent the node with low voiced space weight from splitting. Another component $C_d(w_1, w_2)$ indicates the effect of the difference of the voiced space weight between N_1 and N_2. By using $C_d(w_1, w_2)$, we prefer to remove the unvoiced data from the node at first.

3.3.2 Considering space weight for pitch generation

For a synthetic state with the duration D_c (in frames), we denote the voiced space weight as W_c. The threshold of UV decision is defined as T_{uv}. The original strategy of the UV decision in the pitch generation can be described as following rules.

a. If $W_c \geq T_{uv}$, all the frames in current state is judged as voiced.
b. If $W_c < T_{uv}$, all the frames in current state is judged as unvoiced.

From the rules, the unvoiced/voiced (UV) decision is made in a hard way by comparing the voiced space weight and the UV threshold. Due to this, there are some inconsistency between the generated spectrum and pitch. Especially when the spectrum

with the voiced characteristic is related to the unvoiced pitch, the quality of the speech synthesized by filter is too bad.

We modified the strategy of the UV decision to a smooth way. W_p and W_s denote that of preceding and succeeding state, respectively. Then the new strategy of the UV decision is:

a. If $W_c \geq T_{uv}$, all the frames in current state are judged as voiced.
b. If $W_c < T_{uv}$ and $W_p + W_c \geq 1$, then the first $D_c * (W_c + W_p - 1)$ frames are judged as voiced.
c. If $W_c < T_{uv}$ and $W_s + W_c \geq 1$, then the last $D_c * (W_c + W_s - 1)$ frames are judged as voiced.
d. The other frames are judged as unvoiced.

By considering the space weight for pitch generation, it is consistent with the tree-based clustering procedure.

3.4 Tree-based clustering

3.4.1 Meaning of questions

In the tree-based clustering process, the tree node was split by the best question regarding to the score. Sometimes, there are several questions with the same best score, especially for the low-level tree node, which due to the limited training data compared to the implausible combination number of the contextual features. For example:

N_PhraseLen<40 vs. N_Phrase==XX

The first question is whether the syllable number of next phrase is less than 40. The second question is whether there is no next phrase. When the syllable number of the next phrase is always less than 40 in the training data of current tree node, these two questions have the same score. In fact, only the second question is the best question, which has the general meaning for any data, whereas the first question is only fit for current training data. If the first question was selected to split the tree node, the decision tree will make wrong decision when the next phrase of the testing data has more than 40 syllables, which will be regarded as no next phrase. We should find a way to judge which question is more suitable or meaningful.

As the current tree-based clustering is a full statistic procedure, there is no parameter represent the meaning of the questions. Also it is very difficult to add some knowledge-based strategy under the current framework. Taking account the influence of this problem, here we only focus to solve some special case related to silence, pause and the boundaries, which will result in fatal errors. Usually, the questions related silence, pause and boundary have exact and general meaning for close and open data. Since the names of the questions were designed in regular, the practical solution for this special case is very simple, just to match the key words in the question names. For example, when several questions have the same score, we search the key words in the names of these questions. If there is matched question, we select it as the best question to split the tree node.

It should be noted that we didn't really solve the problem. We just deal with some special case to avoid the fatal errors. We need more efforts to solve the problem.

3.4.2 Threshold for splitting

There are several threshold adopted for splitting the tree node. One is the threshold of the score to split the tree node. In minimum description length (MDL) criterion, it was calculated automatically from the training data, which is one of the advantage of MDL criterion compared to the ML criterion, where it is set by hand in a heuristic way. Another threshold is the minimum number of training data in the leaf node. In the original clustering algorithm, this threshold was adopted in frame scale. In practical application, we found there are few training samples in some leaf node. By analysis, it is caused by the frame number of some data is quite large. Therefore, we adopted the minimum number of the training samples as the thresholds.

3.5 Duration modeling

3.5.1 HRest for HMM training

In the HMM training procedure, the embedded training was applied for full context HMM and clustered HMM training using HERest tool. In the embedded training, only the transcription is used, which means the time alignment labeled by hand have no direct effect on the duration modeling.

Here, we apply the HRest instead of HERest tool for full context HMM and clustered HMM training. As the difference between the HRest and HERest is just the hard/soft boundary used for path searching in the parameter re-estimation, in practical application we use the HERest tool with the pre-segmented data to realize the function of HRest.

3.5.2 Two-level pause

In the original prosody modeling, the pause has only one level. After applying the model to speech synthesis system, we found the synthetic duration for some small pause is too long, which result in uncomfortable in perception. To avoid this matter, we classified the pause to two levels for duration modeling. For the training data, the pauses were split to two levels regarding to the duration:

Level 1: 80ms ~ 200ms

Level 2: 200ms ~

Also, we add the corresponding questions for tree-based clustering.

4 Experiments and performance

4.1 Experimental condition

The training and testing data consists of 1596 and 84 sentences, and all the data had been hand-labeled. The features used to construct the HMM are including spectrum, f0 and their

delta coefficients. All HMMs were left-to-right models with no skip. Each state was modeled by a single Gaussian distribution with diagonal covariance. The number of state set to 5.

4.2 Performance

4.2.1 Variance floor estimation

In the previous training procedure, the variance floors are set to the same for each parameter. Here, we applied the variance floor estimation for each parameter, and the results are shown in Table II-4 and Table II-5.

Table II-4. Number of the leaf node in the decision tree of each state model

		Baseline		HCompV
Cepstrum	S2	190	0.6%	189
	S3	428	1.3%	486
	S4	444	1.3%	460
	S5	334	1.0%	346
	S6	203	0.6%	200
F0	S2	346	1.0%	278
	S3	1100	3.3%	336
	S4	1387	4.1%	307
	S5	1042	3.1%	380
	S6	728	2.2%	613
Duration		1059	3.2%	817

Table II-5. Objective evaluation results for F0 and duration model

		Baseline	HCompV
F0 (Hz)	Abs. Mean error	22.69	23.56
	RMSE	27.48	27.64
Duration (ms)	Abs. Mean error	33.87	32.85

As can be seen in Table II-4, the number of the leaf node in the F0 decision tree became more reasonable regarding the number of the training data. Furthermore, the generated pitch contour was improved from the subject perception. Almost all of the obvious pitch errors have been eliminated. However, we can not find the improvement from the objective evaluation results of F0 values, but a little bit deterioration. In this point of view, the objective evaluation is not suitable for prosody modeling. In the next experiments, we focused on the perception results.

4.2.2 Considering space weight

Firstly, we consider the space weight for tree-based clustering. Different combination of the values of α, β and T_{uv} have been tested, and α, β and T_{uv} set to 4, 1 and 0.5 finally. From the generated decision tree, the first question to split the root node change to

"C_Voiced", which means whether current half-syllable is voiced. Also, we can find other similar question in the high level of the F0 decision tree. From this point of view, the generated decision tree is more reasonable. Also the generated F0 was improved from the perception results. The major improvement is on the head and tail of the voiced unit, where the spectrum and the F0 are more consistent.

Next, we consider the space weight for pitch generation algorithm. To be consistent with the tree-based clustering, T_{uv} set to the same value – 0.5. From the perception result, the speech errors due to the inconsistency between generated spectrum and F0 are almost eliminated.

4.2.3 HRest for HMM training

Using HRest instead of HERest for full context and clustered HMM training. The results of generated duration (not including pause and silence) are shown in Table II-6.

Table II-6. Effect of training strategy on duration modeling

	Mean error	Abs. Mean error
HERest	-12.88	32.85
HRest	-12.22	28.16

From the table, the absolute mean errors between the generated duration and the original duration were reduced. But, there is no any improvement on perception, but a little bit deterioration. So we didn't adopt this training strategy in the final prosody modeling.

4.2.4 Meaning of questions

In our system, the key words used to choose the best question from the questions with the same best score are including "XX", "Sil" and "Pau". After applying key-word-matching strategy, most questions used to split the tree node have the correct and general meaning, especially at the high level of the decision tree. However, the problem is not completely solved. We found another phenomenon, which is that the score of the real best question is a little bit lower than the "best" question. By analysis, the difference between these two questions is only one or two related data. Therefore, we modified the strategy to multiply a certain weight to the score of the question with key word matching. The weight adopted here is 1.05. Finally, almost all the questions on the high level of the decision tree have the correct and general meaning.

4.2.5 Final performance

In the final system, we applied the above techniques, including variance floor estimation, considering space weight for clustering and pitch generation, and considering the meaning of question by key word matching. Also, we used the threshold of the training sample in the leaf node as stopping criterion, and set it to 15. For duration modeling, the two-level pause was used. From the perception results of the final system, it sounds quite natural.

5 Summary

We applied the HMM-based prosody modeling for Chinese application. The contextual feature and the question set were designed by considering the Chinese characteristics. Also, we improved the tree-based clustering technique by considering the space weight and the real meaning of the question to calculate the splitting score. From the perception results of the final system, the synthetic prosody sounds quite natural.

III. Automatic detection of Japanese vowel devoicing

1 Background

Corpus-based speech synthesis has been popularly used due to its high quality, which critically depends on the accuracy and the quality of the speech corpus. In corpus construction [25], phonetic transcription is generally performed according to a pronunciation dictionary. However, the high vowels in Japanese, especially in the Tokyo dialect, are often but not always devoiced in real speech when they are in some particular phonetic environments. This phenomenon is customarily referred to as high vowel devoicing, and considered as a vowel deletion in perception. Many researchers with phonological or phonetics backgrounds have analyzed this phenomenon from different perspectives [26][27][28]. In addition to the phonetic environment, vowel devoicing is also affected by some other factors, including position in the utterance, pitch accent location, rate of speech, etc [27]. As vowel devoicing can not be exactly judged from text by using rules, we should detect it according to the characteristics of the speech data.

In this paper, the detection of vowel devoicing can be regarded as a recognition task, simply classifying the high vowel unit as a voiced or devoiced unit. Due to this, the conventional HMM-based method is applied, and two kinds of likelihood differences are adopted as voicing measures for different focuses. As the discriminative training method has been successful on speech recognition [5][6], we apply it to voiced/devoiced HMMs training to improve performance. Usually, the acoustic features that are used to construct the HMM includes spectrum and energy. Taking into account the dimension of the parameters, the energy parameter has a very limited effect compared to the spectrum parameter. In fact, it should have more effect on the detection of the devoiced units. Also, there are some other features that can discriminate voiced/ devoiced units, including autocorrelation (AC) and duration. Due to this, we also try to incorporate these voicing features in order to improve the detection accuracy.

The rest is organized as follows. In section 2, we introduce the HMM-based method, where two likelihood differences were adopted as voicing measure for different detection focuses, and the discriminative training method is applied to voiced/devoiced HMM training. In section 3, we incorporate the voicing features, including duration, energy and autocorrelation (AC), and combine them with the likelihood differences in several different ways. Finally, the experiments are performed in section 4 to examine the effect of discriminative training and the voicing features, and a discussion is presented in section 5.

2 HMM-based method

Since the detection of vowel devoicing can be regarded as a recognition task, simply classifying the high vowel unit as a voiced or devoiced unit, the HMM-based method was applied to this task. For each high vowel, two HMMs, a voiced and a devoiced HMM, are

constructed, and trained with human-labeled data. In the recognition process, it is performed to construct the recognition network by fixing all the units except for the high vowel unit in the original transcription, and search for the optimal path with the maximum likelihood.

2.1 Two likelihood differences

In the real task, we have different detection focuses for different applications, e.g. focus on removing the devoiced unit from the corpus. However, the current HMM-based framework does not provide a mechanism to detect the devoicing unit for different focuses. Therefore, we tried to extract the implied voicing measures from the recognition process.

For a high vowel unit in an utterance, let's consider the likelihood of three typical transcriptions. The first one is the transcription where the high vowel unit is transcribed as a voiced unit. Accordingly, the second one is the transcription with a devoiced unit. The last one is the transcription where the vowel unit is deleted. We then get two likelihood differences as follows:

$$\Delta L_{norm} = L_{devoiced} - L_{voiced}, \tag{III-1}$$

$$\Delta L_{tee} = L_{deleted} - L_{voiced}, \tag{III-2}$$

where $L_{devoiced}$, L_{voiced} and $L_{deleted}$ are the likelihood of the utterance where the high vowel unit is transcribed as a devoiced, voiced or deleted vowel unit, respectively. With these two likelihood differences, it is obvious that the current HMM-based method can be represented as the following rules:

R1. For a normal HMM, the unit is detected as a devoiced unit when $\Delta L_{norm} > 0$, otherwise it is detected as a voiced unit.

R2. For a tee-model HMM, the unit is detected as a devoiced unit when $\Delta L_{norm} < 0$ or $\Delta L_{tee} < 0$, otherwise it is detected as a voiced unit.

In fact, these two likelihood differences are more general than current HMM-based framework. We can design the rules as follows:

R3. When $\Delta L_{norm} > L_{thr}$, the unit is detected as a devoiced unit, otherwise it is detected as a voiced unit.

R4. When $\Delta L_{norm} < L_{thr}$ or $\Delta L_{tee} < L_{thr}$, the unit is detected as a devoiced unit, otherwise it is detected as the voiced unit.

where L_{thr} is the threshold. These rules show that the current HMM-based framework is just a special case with the threshold equal to zero. In rule R3 or R4, the unit is detected as a devoiced unit if the related likelihood difference is larger than the threshold, otherwise it is detected as a voiced unit. From this point of view, the likelihood difference can be regarded as a voicing measure. Moreover, we can set different thresholds in the rules for different detection focuses. For example, if we want to place a greater focus on removing the devoiced unit from the corpus, we can set the threshold to less than zero.

In addition, we examined the effect of these two likelihood differences by other rules, which are as following:

R5. When $\Delta L_{tee} > L_{thr}$, the unit is detected as a devoiced unit, otherwise it is detected

as a voiced unit.

R6. When $w_{norm} * \Delta L_{norm} + w_{tee} * \Delta L_{tee} > L_{thr}$, a unit is detected as the devoiced unit, otherwise it is detected as the voiced unit.

where w_{norm} and w_{tee} are the related weights.

2.2 Discriminative training for voiced/devoiced HMM training

The discriminative training method and the Minimum Classification Error (MCE) criteria based on the Generalized Probabilistic Descent (GPD) framework have been successful in training HMMs for speech recognition [6]. Since the HMM-based detection of vowel devoicing is a recognition task, we applied the discriminative training method for the voiced/devoiced HMM training.

As the detection of devoicing unit can be regarded as a simply recognition task, the loss function is designed similarly. In this task, it only needs to discriminate the voiced and devoiced HMMs in this task, then the error measurement is simply defined as the likelihood difference between correct and incorrect transcription, that is

$$d(X,\Lambda) = g_i(X,\Lambda) - g_c(X,\Lambda), \tag{III-3}$$

where $g_c(X,\Lambda)$ and $g_i(X,\Lambda)$ are the likelihood of correct and incorrect transcription, X is a feature vector and Λ represents the system parameters. Accordingly, the loss function is defined as

$$\ell(X,\Lambda) = \frac{1}{1+\exp(-\gamma d)}, \tag{III-4}$$

where γ is a positive number. With the loss function definition, the GPD algorithm was performed to minimize the overall expectation loss. The detail can be found in Section I.

Here, the step size is defined as

$$\varepsilon_t = \frac{1}{\alpha + \beta t}, \tag{III-5}$$

where α and β are positive numbers, and t is the index of the training sample. Usually, α and β are set to fixed values for all HMMs. However, the number of devoicing units is much less than the number of voicing units in speech corpus, which means the training data is unbalanced for the voicing and devoicing HMMs. To compensate for this unbalance, we designed different step sizes for the voicing and devoicing HMM training. Here, we define the ratio of the step size between voicing and devoicing HMMs as

$$\eta = \alpha_d / \alpha_v = \beta_d / \beta_v, \tag{III-6}$$

where α_d, β_d are related to the devoicing HMMs, and α_v, β_v are related to the voicing HMMs. In our experiments, we examine the effect of discriminative training with different η values.

3 INCORPORATION OF VOICING FEATURES

In conventional HMM-based methods, only the spectrum and the energy are used to construct the HMMs. Taking into account the dimension of the parameters, the energy parameter has a very limited effect compared to the spectrum parameter. In fact, it should have more effect on detection of devoiced units. Also, there are other features that can discriminate voiced/devoiced units, including duration and AC. Here, we try to incorporate them to improve the detection performance.

3.1 Voicing features

The AC had been used for voicing measure of speech [29], which represents the periodicity of the signal in time domain. For one frame of the signal $x(t)$ with length T, the AC is calculated as

$$R(t) = \frac{1}{T-t} \sum_{\tau=0}^{T-t-1} x(\tau)x(\tau+t) . \tag{III-7}$$

Then the voicing measure f_{AC} is defined as the maximum value of the normalized autocorrelation in the interval of natural pitch periods [2.5ms~12.5ms]:

$$f_{AC} = \max_{2.5ms<t<12.5ms} R(t)/R(0) . \tag{III-8}$$

When f_{AC} is close or equal to 1, it indicates voiced frames. And values of f_{AC} close to 0 indicate the voicedless frames.

In addition to the AC, there are other features that reflect the voicedness of the high vowel unit. Usually, devoiced units have a very short duration even close to zero, whereas voiced units have a long duration. Also, a unit with very low energy can be regarded as a devoiced unit, even though it has a high AC value. As these features, including AC, duration and energy, reflect the voicedness of speech in certain aspects, here we call them voicing features.

Equipped with these voicing features, the next question is how to incorporate them into the current framework. The simplest way is to combine these features with the spectrum parameters to construct the HMMs. From the results of preliminary experiments, that does not work. We thus need to find another suitable way to incorporate these features.

3.2 Combining voicing features and likelihood differences

Here, we incorporate the voicing features in a post-processing procedure. The whole procedure is performed as follows:

a. The HMM-based method is used to perform devoicing detection, and the two likelihood differences are recorded.

b. Based on the alignment of the first step, the voicing features are calculated for each high vowel unit.

c. The voicing features and the likelihood differences are combined in a certain method to judge whether the high vowel unit is devoiced or voiced.

In this procedure, the key point is the method for combining the voicing features and the likelihood differences. Several different methods have been tested in our experiments.

The first method is to directly sum them up by multiplying the corresponding weights, i.e.,

$$S = w_{ener} f_{ener} + w_{dur} f_{dur} + w_{ac} f_{ac} + w_{norm} \Delta L_{norm} + w_{tee} \Delta L_{tee}. \quad \text{(III-9)}$$

where f_{ener}, f_{dur} and f_{ac} are the energy, duration and AC, ΔL_{norm} and ΔL_{tee} are the likelihood differences, and w_{ener}, w_{dur}, w_{ac}, w_{norm} and w_{tee} are the related weights. With the cumulated score, the rule similar to (R5) is adopted to detect the devoiced unit with certain threshold.

In the second method, we first extracted statistical information, including mean and variance, for each voicing feature from the training data. With the statistical information, the voicing score for each feature f is calculated as:

$$s = (f - m_d)^{\sigma} / v_d^{\sigma} - (f - m_v)^{\sigma} / v_v^{\sigma}, \quad \text{(III-10)}$$

where σ is a positive number, m_d and v_d are the statistical mean and variance of the devoiced unit, and m_v, v_v are related to the voiced unit. Then these voicing score and the likelihood difference are cumulated with the related weights

$$S = w_{ener} s_{ener} + w_{dur} s_{dur} + w_{ac} s_{ac} + w_{norm} \Delta L_{norm} + w_{tee} \Delta L_{tee}. \quad \text{(III-11)}$$

The rule used for devoicing detection is similar to the first way.

As the Classification and Regression Tree (CART) is an effective method for the classification problem, we use these voicing features and the likelihood difference to build a CART tree. By setting different weights for the voiced/devoiced training data, decision trees are built for different detection focuses. In the tree construction, the cross-validating technique was applied to build the right-size tree.

4 Experiments

4.1 Experimental condition

The training data consisted of 2,263 phonetically balanced sentences, including 37,256 voicing units and 3,357 devoicing units. The testing data consisted of 503 sentences, including 9,574 voicing units and devoicing units. All of the data had been hand-labeled. In the HMM-based method, monophone HMMs were used, and the numbers of states and mixture components were three and five for each phoneme, respectively. All HMMs were left-to-right models with no skips. The acoustic features were 16-order MFCC and energy with the delta coefficients. The analysis window size and shift were 20 ms and 5 ms, respectively.

In Japanese, most devoicing phenomena occur in the vowels /i/ and /u/ (related devoiced vowels are transcribed as /I/ and /U/). Furthermore, the devoicing characteristic is vowel-dependent. Due to these factor, the effect of each technique was examined for /i/

and /u/ respectively. In our experiments, receiver operating characteristic (ROC) curves are drawn to represent the evaluation results. See Figure 1 for a schematic representation of an ROC curve. ROC curves are always upward concave. The further the curve extends to upper left corner, the better the measure is for voicing. The definition of the hit and false alarm rate is shown in Table III-1.

Table III-1. Definition of hit and false alarm rate

	Detect as devoiced	Detect as voiced
Devoiced	Hit	False rejection
Voiced	False alarm	Correct rejection

4.2 Likelihood difference for voicing measure

The effect of the two likelihood differences was examined by testing these rules (R3, R4, R5 and R6), and the results for vowels /i/ and /u/ are shown in Figure III-1. Here, rule R3 was regarded as the baseline. As can be seen from Figure III-1(a), the effect of each individual likelihood difference, i.e., rules R3 and R5, are not bad. After cumulating these two likelihood differences with equal weights, the performance of rule R6 is the best. But Figure III-1(b) shows that the cumulation of these two likelihood difference is not the best voicing measure for vowel /u/, which is caused by the bad effect of the likelihood difference ΔL_{tee}, whereas the baseline, i.e., rule R3, has the best performance. This phenomenon shows that the devoicing characteristic is vowel- dependent, and we need to adopt a suitable voicing measure for each vowel.

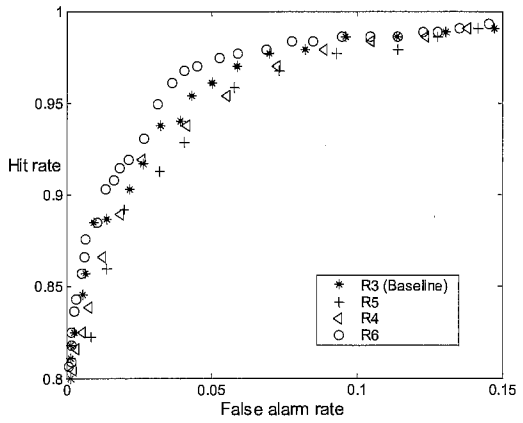
4.3 Effect of discriminative training

We examined the effect of discriminative training with different updating ratios (η=1, 5, 15) between voiced and devoiced HMMs, and the results on vowels /i/ and /u/ are shown in Figure III-2. From Figure III-2(a), the performance of devoicing detection for vowel /i/ was improved to a certain degree after applying the discriminative training. But a different result can be found in Figure III-2(b), where the discriminative training had no effect on improving the detection accuracy of the devoiced unit for vowel /u/. We also see that the performances of the discriminative training with different η value are similar.

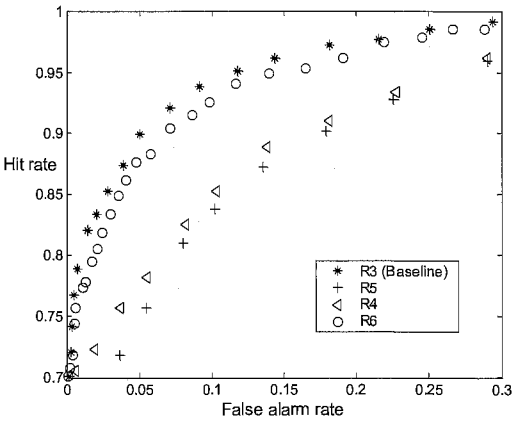
4.4 Effect of voicing features

Finally, we incorporated the voicing features and the likelihood differences in several methods, and the results for vowels /i/ and /u/ are shown in Figure III-3. Figure III-3(a) shows that the greatest improvement was achieved by cumulating the voicing features, or the scores of voicing features, and the likelihood differences with the optimized weights. The improvement by using CART tree to combine these features is very limited, which is not as we expected. In Figure III-3(b), the detection performance was improved by cumulating the voicing features and the likelihood differences, but the improvement is very limited. Also, it can be seen that there is no effect on the devoicing detection for the

vowel /u/ by using other two methods.

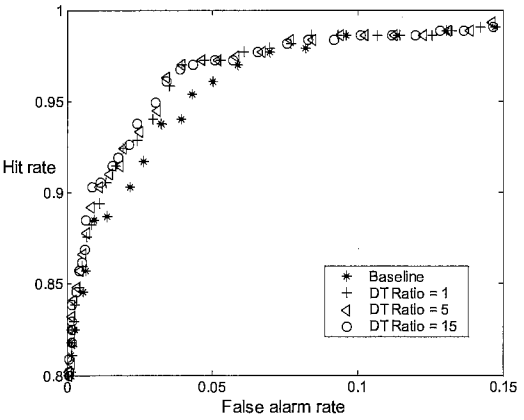


(a) for vowel /i/

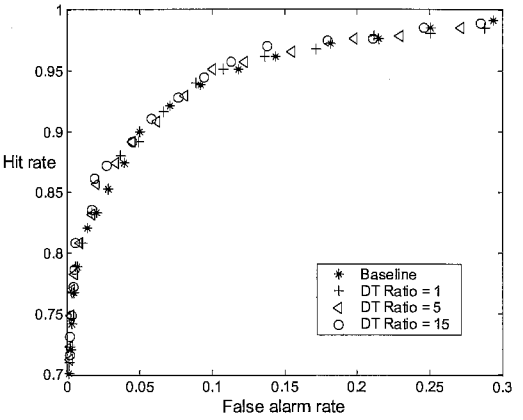


(b) for vowel /u/

Figure III-1. Effect of Likelihood difference

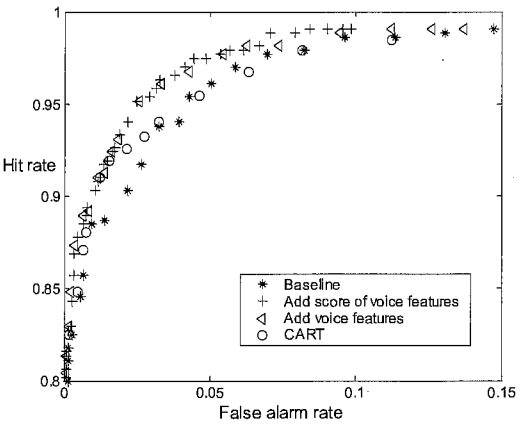


(a) for vowel /i/

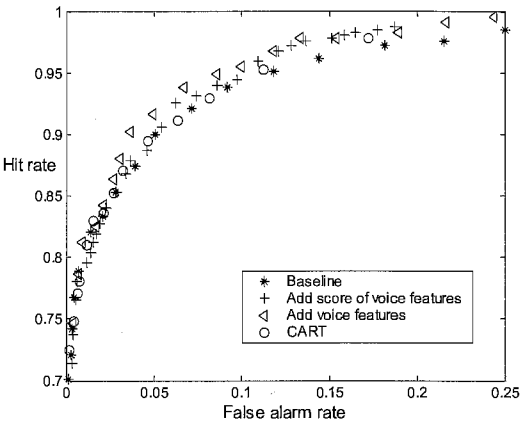


(b) for vowel /u/

Figure III-2. Effect of discriminative training



(a) for vowel /i/



(b) for vowel /u/

Figure III-3. Effect of voicing features

5 Discussion

From these experiments, we can see that the same technique has a different effect on each high vowel. From this point of view, the devoicing characteristic is vowel-dependent, and a different strategy should be designed to detect the devoiced unit for each vowel. Also, the discriminative training has a certain effect on devoicing detection for the vowel /i/, and after incorporating the voicing features in an appropriate way, the performance was improved for both vowels /i/ and /u/. However, there are still some errors, especially for vowel /u/. By analyzing the detection errors, we can see that many errors even cannot be correctly judged by humans. In fact, there is no hard boundary between the devoiced unit and the voiced unit. Taking this into account, the current performance is quite reasonable.

References

[1] P. Carvalho, I. Trancoso and L. Oliveira, "Automatic Segment Alignment for Concatenative Speech Synthesis in Portuguese", in Proc. RECPAD'98 – 10th Portuguese Conference on Pattern Recognition, Lisboa, 1998.

[2] Y.J. Kim and A. Conkie, "Automatic segmentation combining an HMM-based approach and spectral boundary correction", in ICSLP 2002, pp145-148, 2002

[3] A. Sethy, S. Narayanan, "Refined speech segmentation for concatenative speech synthesis", in ICSLP 2002, pp149-153, 2002

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

[5] E. McDermott, "Discriminative training for speech recognition", Dissertation for doctor degree, March 1997

[6] W. Chou, Discriminant-Function-Based Minimum Recognition Error Rate Pattern-Recognition Approach to Speech Recognition, in Proc. IEEE, Vol.88, No.8, pp1201-1223, Aug. 2000

[7] B.H. Juang, W. Chou, and C.H. Lee, "Minimum classification error rate methods for speech recognition", IEEE Trans. Speech Audio Processing, vol.5, pp257-265, May 1997

[8] S.Young, D. Kershaw, J.Odell, D. Ollason, V. Valtchev and P. Woodland, "The HTK Book," Entropic Ltd. 1999

[9] L.R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," Proc. IEEE, vol. 77, pp.257-286, Feb. 1989

[10] S.E. Levinson, "Continuously variable duration hidden Markov models for automatic speech recognition," Comput. Speech Lang., vol. 1, pp. 29-45, 1986

[11] D. Burshtein, "Robust parametric modeling of durations in hidden Markov models," IEEE Trans. Speech Audio Processing, vol. 4, pp.240-242, May 1996

[12] M. J. Russell and R.K. Moore, "Explicit modeling of state occupancy in hidden Markov models for automatic speech recognition," in Proc. Int. Conf. Acoustic, Speech, Signal Processing, pp.5-8, 1985

[13] N.B. Yoma, F.R.McInnes, M.A. Jack, S.D. Stump and L.L. Ling, "On including temporal constraints in Viterbi alignment for speech recognition in noise, IEEE Trans. Speech, Audio Processing, vol.9, 179-182, Feb. 2001

[14] T. Masuko, K.Tokuda, T. Kobayashi and S. Imai, "Speech synthesis from HMMs using dynamic features," Proc. of ICASSP, pp. 389-392, 1996

[15] R.E. Donovan and E.M. Eide, "The IBM trainable speech synthesis system," Proc. of ICASSP, vol. 5, pp. 1703-1706, 1998

[16] M. Plumpe, A. Acero, H.Hon and X. Huang, "HMM-based smoothing for concatenative speech synthesis," Proc of ICSLP, vol. 6, pp. 2751-2754, 1998

[17] T. Yashimura, K. Tokuda, T. Masuko and T.katamura, "Simultaneous modeling of spectrum, pitch and duration in HMM-based speech synthesis," Proc. of Euro Speech, vol. 5, pp. 2347-2350, Sep. 1999

[18] K. Tokuda, T. kabayashi and S. Imai, "Speech parameter generation from HMM using dynamic features," Proc. of ICASSP, pp. 660-663, 1995

[19] T. Fukada, K. Tokuda, T. Kobayashi and S. Imai, "An adaptive algorithm for mel-cepstral analysis of speech," Proc. of ICASSP, vol. 1, pp. 137-140, 1992

[20] H.Kawahara, I. Masuda-Katsuse and A. deCheveigne, "Restructuring speech representations using pitch-adaptive time-frequency smoothing and an instanta-neous-frequency-based F0 extraction: possible role of a repetitive structure in sounds," Speech Communication, vol. 27, no. 3-4, pp. 187-207, 1999

[21] K. Tokuda, T. Masuko, N. Miyazaki and T. Kobayashi, "Multi-space probability distribution HMM", IEICE Trans. Vol.E85-D, No. 3, pp. 455-464, 2002

[22] K. Tokuda, T. Masuko, N. Miyazaki and T. kobayashi, "Hidden Markov models based on Multi-space probability distribution for pitch pattern modeling," Proc. of ICASSP, vol. 1, pp. 229-232, 1999

[23] J. Odell, " The use of context in large vocabulary speech recognition," Dissertation of doctor degree, 1995

[24] J. Rissanen, "Universal coding, information, prediction, and estimation," IEEE Trans. IT, vol. 30, no. 4, pp. 629-636, 1984

[25] A. Ando and E. Miyasaka, "Construction of Japanese News Speech Databases," Proc. Acoustical Society of Japan Spring Meeting, 2-Q-9, Mar. 1997

[26] Kuriyagawa, F., and Sawashima, M. "Word accent, devoicing and duration of vowels in Japanese," Annual Bulletin of the Research Institute of Language Processing pp. 85-108, 1989

[27] Varden John Kevin, "On High Vowel Devoicing in Standard Modern Japanese: Implications for Modern Phonological Theory," Ph. D. dissertation, University of Washington, 1998

[28] Kondo Mariko, "Mechanisms of vowel devoicing in Japanese, " In Proceedings of International Conference on Spoken Language Processing," pp. 61-64, 1994

[29] A. Zolnay, R. Schliiter and H. Ney, "Extraction Methods of Voicing Feature for Robust Speech Recognition," in Eurospeech 2003, pp. 497-500, 2003

[30] Heiga Zen, Jinlin Lu, Jinfu Ni, Keiichi Tokuda and Hisashi Kawai, "HMM-based prosody modeling and synthesis for Japanese and Chinese speech synthesis", ATR technical report, TR-SLT-0032, 2002, Dec.