Reliable Unseen Model Prediction for Vocabulary-Independent Speech Recognition

Sungtak Kim and Hoirin Kim

School of Engineering,
Information & Communications University,
119, Munjiro, Yuseong-gu, Daejeon, 305-714, Korea
{stkim, hrkim}@icu.ac.kr

Abstract. Speech recognition technique is expected to make a great impact on many user interface areas such as toys, mobile phones, PDAs, and home appliances. Those applications basically require robust speech recognition immune to environment and channel noises, but the dialogue pattern used in the interaction with the devices may be relatively simple, that is, an isolated-word type. The drawback of small-vocabulary isolated-word recognizer which is generally used in the applications is that, if target vocabulary needs to be changed, acoustic models should be re-trained for high performance. However, if a phone model-based speech recognition is used with reliable unseen model prediction, we do not need to re-train acoustic models in getting higher performance. In this paper, we propose a few reliable methods for unseen model prediction in flexible vocabulary speech recognition. The first method gives optimal threshold values for stop criteria in decision tree growing, and the second uses an additional condition in the question selection in order to overcome the overbalancing phenomenon in the conventional method. The last proposes two-stage decision trees which in the first stage get more properly trained models and in the second stage build more reliable unseen models. Various vocabulary-independent situations were examined in order to clearly show the effectiveness of the proposed methods. In the experiments, the average word error rates of the proposed methods were reduced by 32.8%, 41.4%, and 44.1% compared to the conventional method, respectively. From the results, we can conclude that the proposed methods are very effective in the unseen model prediction for vocabulary-independent speech recognition.

1 Introduction

The potential application areas using voice interface are enormous. Voice control of consumer devices such as audio/video equipments in home has both commercial potential and well defined functionality that could benefit in user interface. Automotive applications also form a very important area of interest, where the convenience and safety issues play an important role on the choice of the user interface. In addition, user interface using speech recognition is expected to make a great impact on toys, mobile phones, PDAs, and so on. Those applications basically require robust speech

G.I. Webb and Xinghuo Yu (Eds.): AI 2004, LNAI 3339, pp. 599–609, 2004.

recognition immune to environment and channel noises, but the dialogue pattern used in the interaction with the devices will be relatively simple, that is, an isolated-word type. The most straightforward way to implement small vocabulary isolated-word recognizers, which seem to be widely used in the practical applications, is to use speaker-dependent technology. However, training a specific user's speech before the real use could be too inconvenient. Hence, speaker-independent technology is often used, especially when the vocabulary size increases. Even though we may use speaker-independent technology, we cannot avoid changing the target vocabulary occasionally, for example, adding new words or replacing the recognition vocabulary. In this case, we usually have to re-train acoustic models in order to achieve high performance. But, if a reliable unseen model prediction is possible, we do not need to retrain acoustic models to get higher performance.

In this paper, we propose a few reliable methods based on modified binary decision trees for unseen model prediction. Many recognition systems have used binary decision trees for state tying and unseen model prediction. Binary decision trees with splitting questions attached to each node provide an easy representation to interpret and predict the structures of given data set[1]. For more accurate tree-based state tying and unseen model prediction, several factors such as stop criteria, question sets, and question selection in each node should be considered. Of these factors, our approaches focus on stop criterion and question selection, and then we device a new hybrid construction scheme for decision tree combining two approaches. For the stop criterion, we tried to determine an optimal threshold value which allows getting a proper tree size for state tying and unseen model prediction. For the question selection, we added a new condition that enables candidate question to use sufficient training data and to guarantee higher log-likelihood on YES nodes. By using two-stage scheme for decision trees, firstly we can get fairly trained models, and then make the models more effective in the aspects of state tying and more efficient in the aspects of unseen model prediction.

In Chapter II, we briefly review the state tying process based on decision tree. In Chapter III, we present the three proposed methods for accurate unseen model prediction and state tying. Then, the baseline system, the experiments and results are given in Chapter IV and Chapter V. Finally, in Chapter VI, we summarize this work and present ideas for future work.

2 Decision Tree-Based State Tying

Although many other split criteria could be used in decision trees, most of decision tree-based state tying algorithms have used two fundamental criteria, which are likelihood and entropy criteria [2],[3],[4]. The similar probability distributions have to be shared or merged since the basic aim of tree-based state tying is to reduce the number of model parameters and to make the shared parameters more robust. Therefore, the triphone states, whose estimated probability distributions are close to each other in a viewpoint of a distance measure, are tied together. In this paper, we use a log-likelihood gain as the distance measure. The log-likelihood gain is obtained by using the following equation [5].

$$
\begin{aligned}
G(A,B) &= (LL(A)+LL(B)) - LL(AB) \\
&= \frac{1}{2}\left(n_A \log \frac{\sum_{d=1}^{D} \sigma_{d,AB}^2}{\sum_{d=1}^{D} \sigma_{d,A}^2} + n_B \log \frac{\sum_{d=1}^{D} \sigma_{d,AB}^2}{\sum_{d=1}^{D} \sigma_{d,B}^2} \right)
\end{aligned} \tag{1}
$$

Here *AB* is the parent node of nodes *A* and *B* in the binary decision tree, therefore *A* and *B* are the child nodes of the parent node *AB*. n_X is the number of training vectors assigned to node *X*, and $\sigma_{d,X}$ is the variance of component *d* of node *X*.

The formula for the log-likelihood gain can be easily rewritten in a form which only contains sum and squared sum of the observation vector components together with the observation counts. Therefore the equations for computing means and variances of training vectors in each node can be expressed as

$$
\tilde{\mu}_{d,X} = \frac{1}{n_X} \sum_{s \in X} n_s \mu_{d,s} \tag{2}
$$

$$
n_X = \sum_{s \in X} n_s \tag{3}
$$

$$
\tilde{\sigma}_{d,X}^2 = \frac{1}{n_X}\left(\sum_{s \in X} n_s \sigma_{d,s}^2 + \sum_{s \in X} n_s \mu_{d,s}^2 \right) - \left(\tilde{\mu}_{d,X}\right)^2 \tag{4}
$$

where *s* is a state index, $\tilde{\mu}_{d,X}$ is the mean of component *d* of node *X*, $\mu_{d,s}$ is the mean of component *d* of state *s*, and n_s is the number of training vectors in the state *s*. $\tilde{\sigma}_{d,X}^2$ and $\sigma_{d,s}^2$ are the variances of component *d* of the node *X* and the state *s*, respectively. By means of these equations, the re-training computation for each tree construction can be simplified.

3 The Proposed Methods for Unseen Model Prediction

3.1 Modified Stop Decision (MSD) for Optimal Tree Growing

In tree-based unseen model prediction, the tree size becomes a very important factor deciding the accuracy of the predicted models. As the size of tree is larger, the tree has finer resolution due to many leaf nodes. And, if there are many unseen models to be predicted, it is desirable for the tree to get fine resolution. On the other hand, if there are many seen models in the state pool of root node, the probability of observing unseen models will be low. At that time, the size of trees may be reduced because decision trees do not need to get fine resolution.

There is another stop criterion, using minimum number of training vectors of node [6]. In a viewpoint of unseen model prediction, the criterion is not proper because it does not consider whether the probability of observing unseen models is low or not. To overcome those defects, we propose a method that determines optimal threshold value for stop criteria. The method reflects the probability of observing unseen models on the threshold. Then, the threshold values will make trees to get optimal size for more accurate unseen model prediction. A new function for determining optimal threshold values in the state pool of each tree is defined as

$$Threshold = \eta \times N_{seen} \times LL_{norm} \tag{5}$$

$$\begin{aligned} LL_{norm} &= \frac{\text{Log - likelihood of state pool}}{\text{Number of feature vectors in state pool}} \\ &= -\frac{1}{2}\left(D\log 2\pi + \log \sum_{d=1}^{D} \sigma_d^2 + D \right) \end{aligned} \tag{6}$$

where η is a weighting factor to control the number of tied sates, N_{seen} is the number of seen models in the state pool, LL_{norm} is the normalized log-likelihood of the training vectors in the state pool, D is the dimension of feature vectors, and σ_d^2 is the variance of component d of feature vectors. In Eq. (5), N_{seen} controls the threshold value for reliable unseen model prediction. That is, as N_{seen} is larger, the threshold value becomes higher. This is motivated from the fact that, if there are many seen models in the state pool, the probability of observing unseen models to be predicted will be smaller and the tree does not need to get fine resolution for unseen model prediction. On the other hand, if N_{seen} is small, the threshold value must be lower, since the probability of observing unseen models will be higher and the tree needs to get high resolution. On the other hand, LL_{norm} is determined by the variance of feature vectors in the state pool, and it controls the threshold value in the aspect of state tying. That is, as the variance in the state pool becomes larger, the threshold value will be lower. If the state pool has a larger variance, the decision trees should have lots of tied states as possible. This is reasonable for robust state tying because, as the variance of state pool is larger, we need a larger tree and the threshold value must be lower. In conclusion, N_{seen} and LL_{norm} mutually compensate for reliable unseen model prediction and state tying.

3.2 Reliable Question Selection (RQS) Focused on the YES Node

In the tree-based state tying with the likelihood-based framework, the common criterion[5],[7],[8] of choosing a question is formulated as

$$Q^* = \arg\max_{Q}(G(A,B)) \tag{7}$$

where G(A,B) is the log-likelihood gain in node AB. G(A,B) is expressed in Eq. (1). The drawback of using Eq. (7) is that this does not guarantee sufficient training vectors nor higher log-likelihood in the YES node even though the chosen question has the maximum log-likelihood gain. In binary tree-based unseen model prediction, the YES node is more important than the No node because the YES node reflects the context effect of the question itself better than that of the NO node. From the fact, it seems to be desirable that we choose the question providing sufficient training vectors to the YES node and having higher log-likelihood in the node for accurate unseen model prediction.

Thus we propose a new criterion of choosing the question to provide sufficient training vectors and higher log likelihood to the YES node as follows.

$$Q^* = \arg\max_{Q}(G(A,B)) \quad if \;\; n_A > n_{ave} \tag{8}$$

$$n_{ave} = \frac{1}{N}\left(\sum_{n_s \leq 2\sigma} n_s\right) \tag{9}$$

where n_A is the number of training vectors in the YES node, n_{ave} is the average number of the training vectors for states in the state pool, and N is the total number of states in the state pool. In Eq. (9), n_{ave} is the number of training vectors in states that are included in the confidence interval 95%. Here we assume that the number of the training vectors in states has a Gaussian distribution. So, we do not use too large or too small number of training vectors in states to get n_{ave}.

By using this technique, decision trees can guarantee that YES nodes have a sufficient number of training vectors and higher log-likelihood. In result, we can precisely predict unseen models by using the reliable question selection.

3.3 Two-Stage Decision Tree (TSDT) Combining the RQS and MSD Algorithms

When we use triphone models as acoustic models, it is very difficult to construct a training database so that all the possible triphone models have a similar number of training vectors in each model. If the numbers of training vectors in models are significantly different from one another, the decision tree would not be constructed precisely because the likelihood of models may be very sensitive to how many models are trained as shown in Eq. (1) and Eq. (7). In other words, as models are better trained, the likelihood of models is higher. Thus the decision tree using the likelihood framework will be too much dependent on the given training database. In the result, the accuracy of the unseen model prediction for vocabulary-independent speech recognition will be degraded severely.

To overcome this problem, we propose a method that the state of each model has the same number of training vectors as possible. We design a two-stage decision tree: at first we generate a tree from the RQS method with zero threshold value, and next we construct the final tree from the MDS method. In stage 1, we construct a decision tree with RQS method, and assign the probability distribution of shared state in each leaf node to the probability distribution of original states in the leaf node. So, the original models become evenly trained models. Finally at stage 2 we can construct the decision tree which is less dependent on the training database. The two-stage decision tree algorithm is summarized as follows (see also Figure 1).

TSDT Algorithm

- Step 1: Cluster states of seen models by using the RQS method with the threshold zero.
- Step 2: Assign fairly trained states to original states in each leaf node.
- Step 3: Reconstruct thes decision trees by using the MSD method.

Thus Step 1 and Step 2 construct evenly trained acoustic models. Step 3 makes reliable tied models for unseen model prediction by using models sufficiently trained from previous steps.

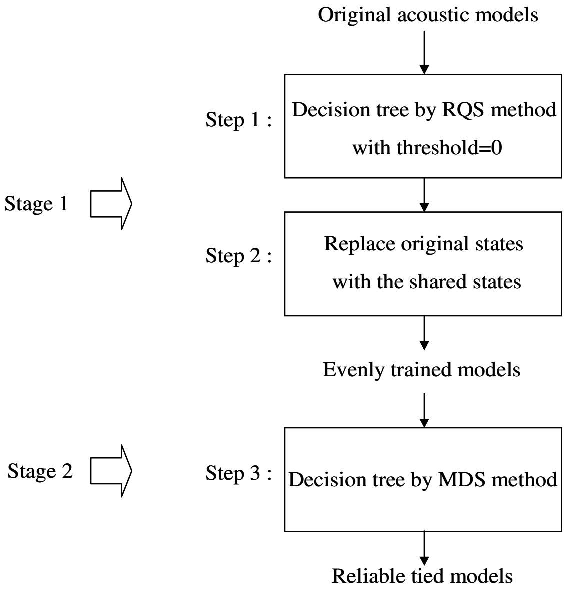
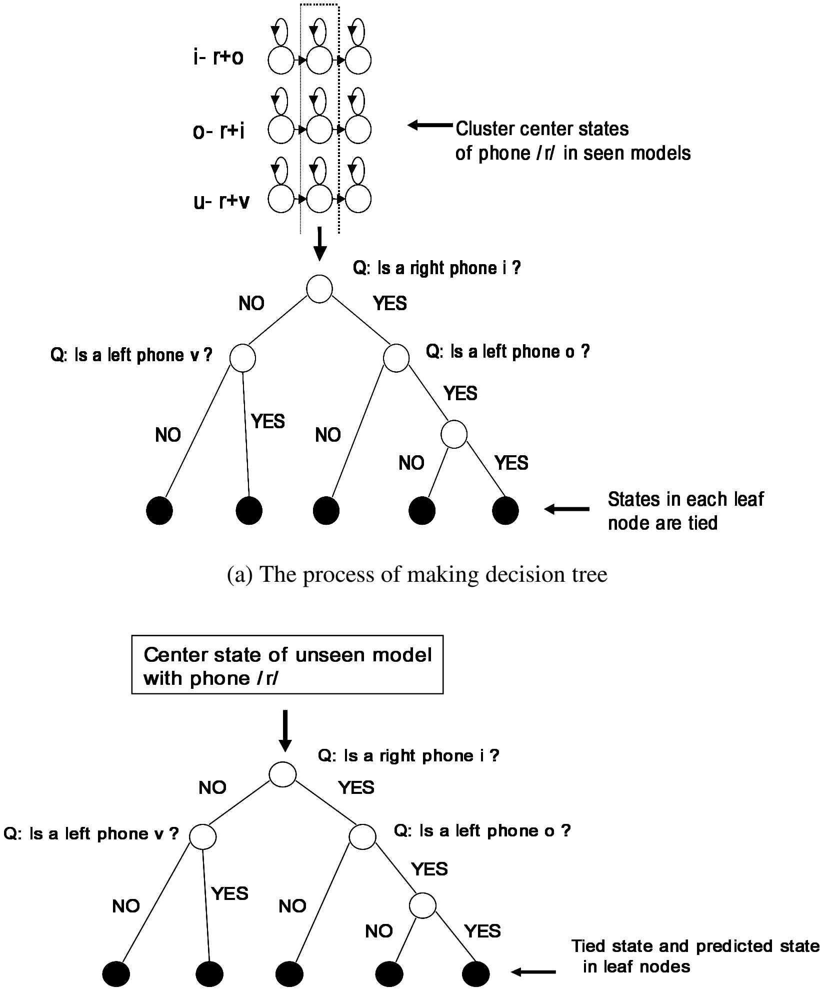

Original acoustic models
Step 1 :
Decision tree by RQS method with threshold=0
Stage 1
Step 2 :
Replace original states with the shared states
Evenly trained models
Stage 2
Step 3 :
Decision tree by MDS method
Reliable tied models

Fig. 1. The process of the TSDT method

4 Baseline System

At first, the input speech is pre-emphasized using the first order FIR filter with a coefficient of 0.97. The samples are blocked into overlapping frames of 20ms and each frame is shifted at a rate of 10ms. Each frame is windowed with the Hamming window. Every frame is characterized by the total 39th order feature vectors. The feature vectors are composed of 13 mel frequency cepstral coefficients (MFCC), their first-order temporal regression coefficients (ΔMFCC), and their second-order temporal regression coefficients (ΔΔ MFCC). Hidden Markov model-based triphones are trained with 3 states left-to-right structure for acoustic modeling.

One decision tree is constructed for every states of each center phone, and all triphone models with the same center phone are clustered into the corresponding root node according to the state position. To get tied states, a decision tree is built using a top-down procedure starting from the root node of the tree. Each node is split according to the phonetic question that results in maximum increase in the likelihood on the train data from Eq. (1). Different phone questions have been investigated in [9],[10], but we have used only simple phone questions because the focus in this work is not on those variations. The likelihood gain due to a node split can be calculated efficiently from pre-calculated statistics of the reconstructed states by using Eq. (2), Eq(3), and Eq (4). The process is repeated until the likelihood gain falls below a threshold. In baseline system, we used a same threshold for each decision tree. After this process is done, states reaching the same leaf node of each decision tree are regarded as similar and so tied. Fig. 2(a) shows this procedure. The resulting clusters of tied states are then retrained and multiple-mixture Gaussian distribution HMMs are estimated.



(a) The process of making decision tree

(b) The process of predicting the unseen models

Fig. 2. An example of decision tree structure in case of center states

When unseen models are observed due to new words added to the vocabulary in-recognition process, the unseen models are predicted by answering to the phonetic questions which already determined in training process and traversing the decision tree from the root node to a final leaf node as shown in Fig. 2(b). The most similar leaf node determined by the decision tree is used as the unseen models.

5 Evaluation

5.1 Speech Data

Speech database used in this work is composed of the PBW(Phonetically Balanced Words) 452 DB and the FOW(Frequently Observed Words) 2000 DB. The PBW 452 DB consists of 452 isolated Korean words, each of which is uttered twice by 70 speakers including 38 males and 32 females. The FOW 2000 DB consists of 2,000 isolated Korean words, each of which is spoken once by two speakers including one male and one female. The FOW 2000 DB includes all the 452 words that are vocabulary of the PBW 452 DB, and the other 1,548 words are different from the words of the PBW 452 DB. These speech data are sampled at 16 kHz and quantized in 16 bit resolution. We used the PBW 452 DB for training and the FOW 2000 DB for test.

For various experiments of vocabulary-independent speech recognition, we established four different test situations from FOW 2000 DB as follow.

- Case 1: The test vocabulary is totally different from the training vocabulary.
- Case 2: The test vocabulary is different from the training vocabulary by 75 %.
- Case 3: The test vocabulary is different from the training vocabulary by 50 %.
- Case 4: The test vocabulary is different from the training vocabulary by 25 %.

The number of words in the test vocabulary for each case is shown in Table 1 where the number of distinct words in the training vocabulary is 452.

Table 1. Number of test words in each case

	S	D	Total distinct words
Case 1	0	1,548	1,548
Case 2	452	1,356	1,808
Case 3	452	452	904
Case 4	452	151	603

In Table 1, S represents the number of words which are the same as the training vocabulary, D represents the number of words which are different from the training vocabulary.

5.2 Performance Comparison of the Conventional Algorithm and the Proposed Methods

In this experiment, we compared the performances of the baseline algorithm and the proposed methods. The baseline algorithm gets a number of tied states according to the same log-likelihood gain values as the threshold values of trees, and the proposed methods get the number of tied states according to the control factor η in Eq. (5).

Table 2. Word recognition accuracies(%) of the baseline and the proposed methods in case 1

# of states	Baseline	MDS	RQS+ MDS	TSDT
1,261	90.31	91.21	91.80	92.18
1,338	89.92	91.21	91.60	91.67
1,397	89.86	91.60	92.18	92.18
1,465	90.05	91.80	91.99	92.25
1,531	89.66	91.67	92.44	92.51

Table 3. Word recognition accuracies(%) of the baseline and the proposed methods in case 2

# of states	Baseline	MDS	RQS+ MDS	TSDT
1,261	92.48	93.03	93.25	93.81
1,338	92.20	92.98	93.58	93.58
1,397	92.37	93.20	94.14	93.92
1,465	92.48	93.36	93.86	93.92
1,531	92.15	93.14	94.08	94.19

Table 4. Word recognition accuracies(%) of the baseline and the proposed methods in case 3

# of states	Baseline	MDS	RQS+ MDS	TSDT
1,261	96.79	97.23	97.23	97.68
1,338	97.35	96.79	97.57	97.68
1,397	97.12	97.01	97.46	97.90
1,465	97.35	97.23	97.35	97.79
1,531	96.90	97.68	97.68	98.23

Table 5. Word recognition accuracies(%) of the baseline and the proposed methods in case 4

# of states	Baseline	MDS	RQS+ MDS	TSDT
1,261	98.84	99.50	99.00	99.34
1,338	99.00	99.50	99.17	99.50
1,397	99.00	99.34	99.34	99.50
1,465	99.34	99.34	99.34	99.67
1,531	99.17	99.34	99.34	99.67

The following tables show the word recognition accuracies of the baseline and the proposed methods in each vocabulary-independent situation. After the tree-based clustering procedure that is based on single Gaussian mixture models, the number of mixture components of all pdfs in all experiments was enlarged to 7 Gaussians per

HMM state. That is, all of the following recognition accuracies were obtained on 7 Gaussians per state.

These results show that the proposed methods have higher or comparable recognition performances when they are compared to the baseline system. Especially, the two-stage decision tree (TSDT) method outperforms other methods in whole cases. To show the effects of the proposed methods in vocabulary-independent speech recognition, the average ERR (Error Reduction Rate) of each case is given in Table 6.

Table 6. Average ERR (%) of the proposed methods

	Case 1	Case 2	Case 3	Case 4
MDS	15.3	10.5	3.0	35.9
RQS+ MDS	20.3	18.9	12.3	17.6
TSDT	21.9	20.2	26.0	50.0

6 Conclusion

In this paper, we proposed three effective methods to construct decision trees for reliable unseen model prediction in vocabulary-independent speech recognition. The MDS method determines the optimal threshold values for accurate state tying and unseen model prediction, the RQS+ MDS method chooses a question guaranteeing sufficient training vectors and higher log-likelihood in the YES nodes, and the TSDT method is a type of model compensation that aligns the new probability distributions to the original models in order to make original ones fairly trained. From experimental results, we could know that these methods were more effective on realistic vocabulary-independent speech recognition corresponding to case 4. The TSDT method was effective on all cases of test environments.

References

1. X. Huang, A. Acero, and H. Hon, "*Spoken Language Processing*," Prentice Hall, 2001.
2. J. Duchateau, K. Demuynck, and D. Van Compernolle, "Novel Node Splitting Criterion in Decision Tree Construction for Semi-Continuous HMMs," in *Proc. of Eurospeech '97*, pp.1183-1186, 1997.
3. S. J. Young, J. J. Odell, and P. C. Woodland, "Tree-Based State Tying for High Accuracy Acoustic Modeling," in *Proc. of Human Language Technology Workshop*, Plainsboro, pp. 307-312, 1994.
4. Mei-Yuh Hwang, Xuedong Huang, and Fileno A. Alleva, "Predicting Unseen Triphone with Senones," *IEEE Trans. Speech and Audio Processing*, Vol. 4, No. 6, pp. 412-419, Nov. 1996.
5. K. Beulen and H. Ney, "Automatic question generation for decision tree based state tying," in *Proc. of ICASSP '98*, pp. 805-808, 1998.

6. Wolfgang Reichl and Wu Chou, “Robust Decision Tree State Tying for Continuous Speech Recognition,” *IEEE Trans. Speech and Audio Processing*, Vol. 8, No. 5, pp. 555-566, Sept. 2000.
7. Daniel Willett, Christoph Neukirchen, J. Rottland, and Gerhard Gigoll, “Refining Tree-based State Clustering by means of Formal Concept Analysis, Balanced Decision Tree and Automatically Generated Model-Sets,” in *Proc. of ICASSP ’99*, Vol. 2, pp. 565-568. 1999.
8. T. Kato, S. Kuroiwa, T. Shimizu, and N. Higuchi, “Efficient mixture Gaussian synthesis for decision tree based state tying,” in *Proc. of ICASSP ’01*, Vol. 1, pp. 493-496, 2001.
9. R. Kuhn, A. Lazarides, Y. Normandin, and J. Brousseau, “Improved decision trees for phnetic modeling,” in *Proc. of ICCASSP ’95*, pp.552-555.
10. A. Lazarides, Y. Normandin, and R. Kuhn, “Improving decision trees for acoustic modeing,” in *Proc. of ICSLP ’96*, pp. 1053-1056.