論文 / 著書情報
Article / Book Information

論題(和文)	
Title(English)	Speech Recognition System in NEC
著者(和文)	篠田 浩一
Authors(English)	Takao Watanabe, Kaichiro Hatazaki, Ken-ichi Iso, Ryosuke Isotani, Koichi Shinoda, Keizaburo Takagi
出典(和文)	, Vol. , No. , pp. 34-46
Citation(English)	Spoken Language Systems, Vol. , No. , pp. 34-46
発行日 / Pub. date	2005, 9

1

Speech Recognition System in NEC

Takao Watanabe
Kaichiro Hatazaki
Ken-ichi Isc
Ryosuke Isotani
Koichi Shinoda
Keizaburo Takag
NEC Corporation

ABSTRACT

NEC Corporation has developed a speaker-independent continuous speech recognition system that uses a demi-syllable speech unit and is applicable to a large vocabulary. To enable high performance speech recognition, high-speed computation methods are used for search processing in the continuous speech recognition and likelihood calculation. Speaker adaptation and environmental adaptation techniques enable robust speech recognition. The methods are used in speech recognition systems for LSIs, telecommunication systems and PCs.

1 Introduction

A speaker-independent continuous speech recognition method, that uses a demi-syllable speech unit[1)] and is applicable to a large vocabulary, has been developed by NEC Corporation.

Speech recognition based on phonetic units, which are smaller than words, is suitable for task-independent or vocabulary-independent speech recognition, as well as for large vocabulary recognition. This is important from the practical viewpoint, because this kind of recognition makes it unnecessary to collect each user's vocabulary utterances or to collect a large number of speakers' vocabulary utterances for each application. Various problems must be solved, though, to realize phonetic unit based speech recognition. Phonetic contextual variations in speech and variations across speakers must be considered. In continuous speech, word boundary variation is also a problem. To limit the computational requirement for both training and recognition, the number of speech units should be small and the speech unit structure simple.

We use a demi-syllable unit as a recognition unit. Because it contains transitional information between phonemes, it can express phonetic contextual variations using

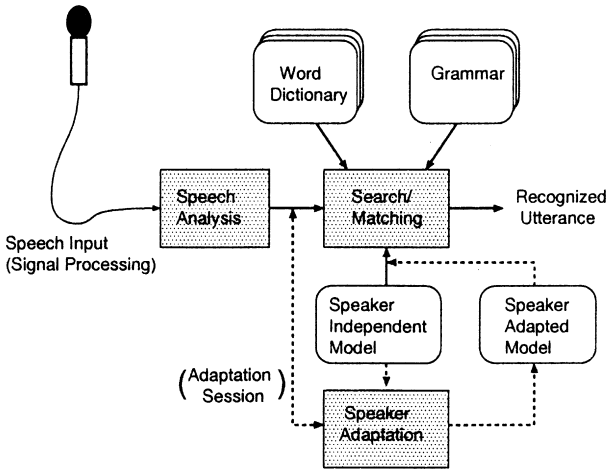

Figure 1: Recognition system

a relatively small number of units. To describe the demi-syllable unit with a small set of parameters, we use a hidden Markov model(HMM) which has a continuous probability density function (pdf) with diagonal covariance.

To achieve high-performance speech recognition, computationally efficient implementation is essential. Several high-speed computation methods have been developed for search processing in continuous speech recognition and likelihood calculation.

From the practical viewpoint, robustness against variance caused by environmental factors, such as background noise, as well as speakers' differences is also necessary, so we have applied speaker adaptation and environmental adaptation techniques.These techniques are effective and computationally efficient.

2 Demi-syllable Unit Based Recognition

2.1 System overview

Figure 1 shows a block-diagram of the recognition system. Speech data is sampled at 16 kHz sampling rate, and analyzed using a 10 msec frame period. As a feature vector for each time frame, mel-cepstral parameters and differential mel-cepstral parameters are calculated from FFT based spectrum.

Models for demi-syllables are trained, using speech data and a training dictionary. Model parameters are estimated using forward-backward training algorithm. In the recognition phase, a recognition dictionary expressed in the form of demi-syllables is used. The demi-syllable models are connected to make a word model, according to the recognition dictionary. Probabilities for the word models are calculated for input speech, and the word which has the largest probability is selected as the recognition result.

2.2 Demi-syllable model

Recognition unit definition is important for speech recognition, using a unit smaller than a word. We use a demi-syllable unit as a recognition unit. The demi-syllable is a half syllable unit divided at the center of the syllable nucleus. The unit has the following desirable properties. (1) The unit has transitional part information, which is important for phoneme recognition, and can treat contextual variation caused by the co-articulatory effect. (2) Unit size is moderate.

Since Japanese syllables have a combined consonant-vowel(CV) structure, the syllable number is relatively small (approximately 120). The demi-syllable can treat both CV transition and vowel-consonant(VC) transition. The latter can not be treated using the syllable. The number of demi-syllable is much smaller than VCV and CVC units.

The unit set basically consists of CV and VC models. Vowels include five Japanese vowels and syllabic nasal[N]. In addition, demi-syllable pairs for vowel-vowel (VV) sequence, geminate consonant (CC) units, and silence units are defined. Recognition units are modeled by the left-to-right model, whose vector output probability is defined by continuous pdfs in each state. A mixture Gaussian pdf is used for the output probability. In order to deal with the variations across speakers for speaker-independent speech recognition, mixture Gaussian pdf HMM is introduced. State numbers are defined for each recognition unit depending on the acoustic structure of the phonemes, where the numbers are determined to be as small as possible. As the model parameter number is relatively small, they should be efficiently estimated using a small amount of training data. Demi-syllables are basically segmented at a consonant start point and a vowel center point. Thus, a CV segment is considered as being from a consonant start point to a vowel center, and a VC segment is considered as being from a vowel center to a consonant start point. Figure 2 shows an example of segments. As the demi-syllable segmentation can be automatically conducted in the training procedure, this segmentation criteria is used for initial segmentation for training. Acoustically similar demi-syllables are merged into one unit class to reduce the unit number. As the VC segment has almost no consonant portion, VC units can be bundled into one unit class based on a consonant category. The consonant for the VC units are classified into nine categories according to manner-of-articulation of the consonant (voiceless stops, voiced stop, and nasal etc.) Allophonic demi-syllable units are prepared for larger variations. The candidates of the allophonic variants are generated according to the following phonological rules: (1) High vowels[i,u] between voiceless consonants may be devoiced. (2) Vowel sequences [ei, ou] may be changed to long vowels.

As training vocabulary, a compact phonetically balanced word set was selected from a Japanese word lexicon. The word set was designed to include all demi-syllables at least once, and to assure that as much allophonic variation as possible should occur. Demi-syllable models are connected into a word model using the training dictionary. The word model parameters are estimated based on forward-backward algorithm using the training data, where all the model parameters in the same demi-syllable

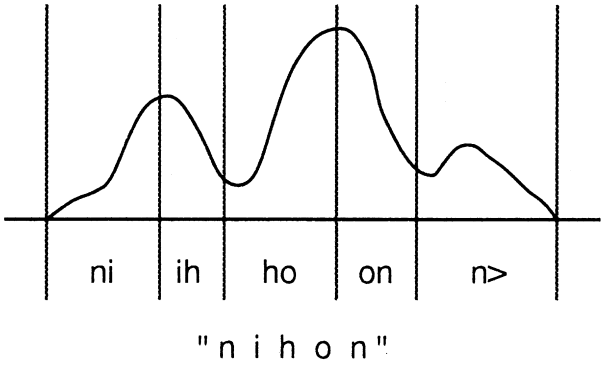

Figure 2: Demi-syllable unit

are tied together. In the training, it is necessary to know the allophonic variant which actually occurred in the training utterances, such as 'devoiced or not'. This is done automatically by determining the variant with the larger probability calculated using the reference speaker's model.

2.3 Syntax network

The finite-state automaton syntax network is used as the grammar in the continuous speech recognition. The syntax network is represented by a directed network composed of arcs representing a word or a subnetwork, and arcs representing the null transition. A subnetwork is again represented by the same kind of directed graph, and is used as the macro representation. The recursive use of a subnetwork is not allowed. The compiler processing develops the syntax network, the word dictionary and the demi-syllable model into a single model network.

In the recognition of continuous speech, the co-articulation effects between words must be considered, as the variation of the speech pattern due to difference of phoneme environment. To handle the co-articulation effect at the word juncture, a demi-syllable model is inserted between words on the syntax network. This enlarges the scale of the syntax network especially when there exist a node with a large number of incoming and outgoing words. However, it is not necessary to insert a word juncture model between each pair of a preceding and a succeeding word. The number of the word juncture models is reduced by grouping the preceding and and the succeeding word sets according to the final vowel and the first consonant, respectively, as is shown in Figure 3. Additionally, applying the bundle processing described below to the word juncture models suppresses the increase of the processing complexity in a large-scale task.

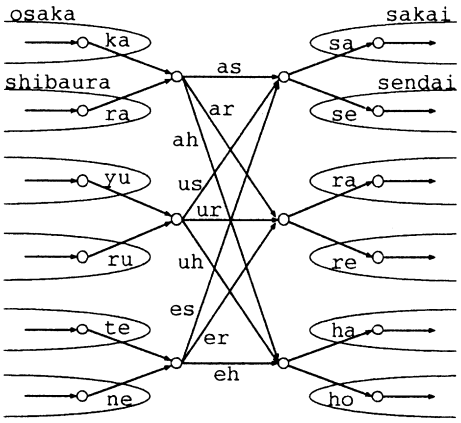

Figure 3: Word juncture model.

2.4 Bundle search

The recognition is executed by determining the optimal path on the developed model network by Viterbi search. To realize the high-speed processing suited to real-time process, the bundle search method is used, which is a high-speed version of the frame synchronous search. The bundle search is the following method for the high-speed processing[2]. When the same word appears at more than one place in the syntax network, the matching for those words are simplified.

In the ordinary frame synchronous method, the matchings between the input pattern and the words at different positions on the syntax network are examined separately, even if the words are the same. Consequently, if a word n appears more than once on the syntax network, the same number of word matching processes as the number of appearances are required for word n. Even if the word is the same, it must be processed as different words, since the initial value for the cumulative distance, which serves as the initial condition for the word matching, depends on the position of its appearance.

In contrast, in the bundle search method, the word matching is executed only once for each word. The principle of the method is described, using the case of Fig.4 as an example. Assume that the word n appears twice ($n^{(1)}$ and $n^{(2)}$ in the figure). The word matching is made only once as follows, for the two positions of appearance with different initial values for the cumulative distance. The minimum of the cumulative distances T_1 and T_2 (let it be T_1 in the figure) is determined, and the word matching is executed using the minimum value as the initial value. The cumulative distance T_1^* is obtained at the end of the word. For the rest positions of appearance, the word matching is not executed. Instead, the difference of the initial values of the cumulative distance ($\Delta = T_2 - T_1$) is used to estimate the cumulative distance after the word matching, ($T_2^* = T_1^* + \Delta$).

By this scheme, the amount of processing is reduced drastically, since the match-

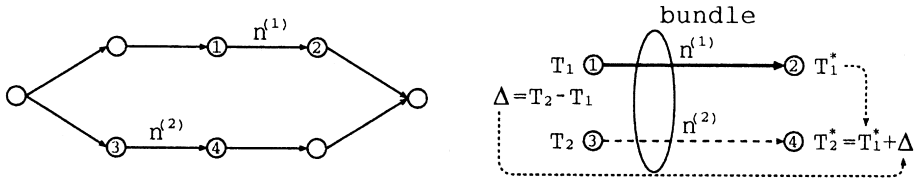

Figure 4: Basic idea of bundle search.

ing process is applied only once for each word. Although the optimality is lost theoretically, all matching paths are retained as an approximation and the quasi-optimal solution is obtained.

The improvements of the speed in the bundle search and the beam search described below are based on different standpoints. The bundle search executes a coarse approximate processing over the whole search space, while the beam search executes the processing only for the search paths with a high possibility. Those two methods can be combined in application.

2.5 Tree-structured lexicon and beam search

For very large vocabulary recognition, it is effective to arrange the phonetic representation for the words in the vocabulary into a tree. The computational amount of recognition is reduced by executing the processing in common for the same recognition units at the beginning portion of many words, compared with ordinary linear arrangement of the lexicon. The tree-structured lexicon is especially effective when the lexicon has a hierarchical structure.

Another common approach to speed up the recognition process is the beam search. In the beam search, only likely hypotheses are retained for further processing and others are pruned.

Usually wider beam width is necessary at the beginning of utterance because scores of hypotheses are uncertain. Using the tree-structured lexicon, however, enables effectively narrowing the beam width at the beginning of utterance.

2.6 High speed speech recognition using tree-structured probability density function

In large vocabulary speech recognition, continuous output probability with mixture density is commonly used. To realize real-time speech recognition system with small computational cost[3], it is important to reduce the amount of likelihood calculation of Gaussian probability densities (*pdfs*), as well as the amount of speech analysis and search (word matching) processing.

Let a sequence of input vectors be $X = \{x_1, ..., x_t, ..., x_T\}$, and let a *pdf* set, composed of the Gaussian *pdfs* for all mixture components, all states and all recognition

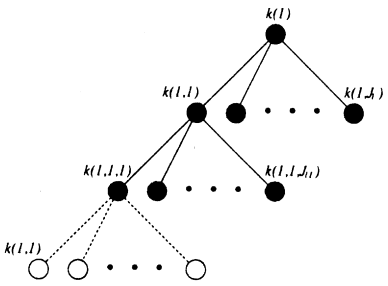

Figure 5: Tree-structure of pdf

units, be $Y = \{N_1[.], ..., N_k[.], ..., N_K[.]\}$. Each element in the *pdf* set is called an *element pdf*. At each time t, in the recognition, it is necessary to calculate a likelihood set

$$B_t = \{N_k[x_t], k = 1, ..., K\}. \tag{1.1}$$

Let's consider how to efficiently obtain the likelihood set B_t. The basic idea considered here is as follows: For the *element pdf* which is likely to correspond to time t (that is, the *element pdf*, whose likelihood $N_k[x_t]$ at time t is large), the likelihood is precisely calculated. For the *element pdf* which is unlikely to correspond to time t, calculation is done coarsely.

To realize the principle mentioned above, the *pdf* set Y is expressed as the form of the tree-structure, and the calculation of Eq. (1.1) is done by searching the tree.

Figure 5 shows the tree-structure. A leaf node of the tree corresponds to an *element pdf*. A non-leaf node corresponds to a cluster composed of *element pdf*s. Each cluster consists of other clusters or *element pdf*s. To each cluster is attached a *cluster pdf*, obtained by approximating the mixture of all *element pdf*s in the cluster by a single Gaussian *pdf*.

In the recognition, the likelihood set B_t is calculated by searching the tree. First, calculate likelihoods from the *cluster pdf*s for nodes $\{k(1,1), ..., k(1,j), ..., k(1,J_1)\}$ in the first stage. Then, select the node which gives the largest likelihood. It is possible to select the multiple nodes by selecting the N-best likelihoods. For the child nodes of the selected nodes, this procedure is repeated. This tree search procedure is continued until all selected nodes reach the leaf nodes.

For the selected leaf nodes, the likelihood is calculated from the *element pdf*s of the nodes. For unselected leaf nodes, the likelihoods are approximated by the likelihoods calculated from the upper node *cluster pdf*s. Using this procedure, the likelihood set B_t in Eq. (1.1) can be calculated very efficiently.

The tree-structured *pdf* is designed beforehand by a top-down clustering technique. First, the *element pdf* set Y is set as the initial *pdf* set for clustering, and the *pdf* set is divided into clusters. The number of the cluster is given beforehand. Next,

each cluster is further divided into sub-clusters. This division procedure is repeated the number of times determined beforehand.

For each cluster, a *cluster pdf* is calculated from the set of the *element pdf*s which belongs to the cluster. This is done by approximating the mixture of the *element pdf*s by a single Gaussian *pdf*, as follows:

$$\mu_m(i) = \frac{1}{K}\sum_k \mu_k(i),$$

$$\sigma_m(i) = \frac{1}{K}[\sum_k \sigma_k(i)^2 + \sum_k \mu_k(i)^2 - K\mu_m(i)^2],$$

where Σ denotes the summation regarding element number k belonging to cluster m, $\mu_k(i)$ and $\mu_m(i)$ are i-th components of the mean vectors for the *element* and *cluster pdf*s, respectively, and $\sigma_k(i)^2$ and $\sigma_m(i)^2$ are i-th diagonal components for the diagonal covariance matrices of the *element* and *cluster pdf*s, respectively.

2.7 Unknown utterance rejection using likelihood normalization

One of the problems posed for the speech user interface is to cope with the utterance out of the vocabulary or grammar, which is the object of recognition of the system. Here we proposed a method of rejection of the unknown utterance based on the likelihood determined by eliminating the likelihood variation due to the speaker and the environment[4]. The basic process is to utilize likelihood obtained in the recognition of the syllable sequence, to normalize the likelihood obtained under the linguistic constraint, and to use the normalized likelihood in the decision of the rejection.

Let the likelihood for the input speech obtained as a result of the syllable recognition without task constraint be P_r, and the likelihood obtained as a result of recognition with task constraint be P_t. One can assume that the two likelihoods differ only slightly for the utterance satisfying the task constraint. When the syllables are recognized correctly, the two likelihoods agree. On the other hand, for the unknown utterance which does not satisfy the task constraint, it is anticipated that the two likelihoods exhibit a greater difference. By those considerations, the difference between the two likelihoods is defined anew as the normalized likelihood L' and the rejection of the unknown utterance is decided by a threshold decision.

$$L' = (L_t - L_r)/I \tag{1.2}$$

where $L_t = \log P_t$, $L_r = \log P_r$, and I is the input length. In the decision about the rejection, the normalized likelihood L' is compared to the predetermined threshold, and it is decided whether the input is rejected as unknown utterance or is accepted.

The interpretation of this method is made as follows. Even if the same syllable sequence is uttered, the speech pattern differs depending on the speaker or environment and the obtained likelihood also varies. Such a variation, however, appears in similar ways in the task recognition and the syllable recognition, and one can anticipate that the effect of the variation is contained in common to the two likelihoods.

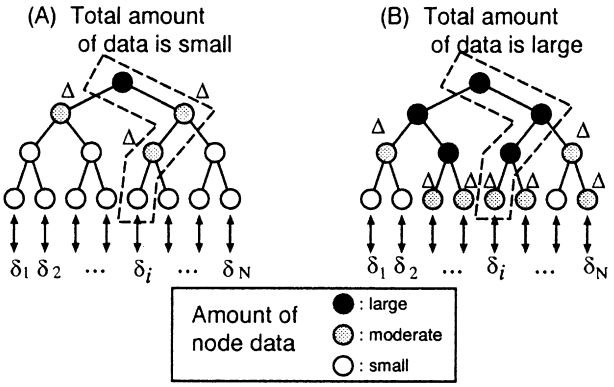

Figure 6: Speaker adaptation using AMCC

By cancelling this effect, the likelihood due to the linguistic constraint will be extracted. The reason for taking the difference between the logarithmic likelihoods is to cancel such an effect of variation.

3 Speaker Adaptation using AMCC

In general, speaker adaptation methods are expected to have the following two properties: 1) the recognition accuracy is improved with even a small amount, of data for adaptation, 2) the recognition accuracy increases as the amount of data increases, even when the amount of data available is large. Speaker adaptation using Autonomous Model Complexity Control (AMCC) [5] is one method that have both properties. In AMCC, the degree of freedom with which free parameters can be estimated is *autonomously* controlled according to the amount of data for adaptation.

Each state in continuous density mixture Gaussian HMMs has an output probability density function, which is the weighted summation of K Gaussian components. In AMCC, SI mean μ_{i0} of Gaussian component i is mapped to the unknown SD mean $\hat{\mu}_i$ as follows:

$$\hat{\mu}_i = \mu_{i0} + \delta_i, \qquad i = 1, \ldots, N \times K,$$

where δ_i is a shift parameter from the SI mean, and N is the number of states in HMMs. As means of controlling the number of free parameters, AMCC uses a tree structure for the Gaussian components(Figure 6). In this tree structure, each leaf node i corresponds to each Gaussian component i, and a tied-shift Δ_j is defined for each non-leaf node j (including a top node). The shift δ_i for each leaf node i is estimated using a training algorithm such as Viterbi algorithm or Forward-Backward algorithm, and then, the tied-shift Δ_j for each non-leaf node j is calculated as a shift shared by all the leaf nodes that fall below the node j. To autonomously control the number of parameters, we set a threshold on the amount of *node data*. Here *node*

data is defined for each node j as the data used for estimating the tied shift Δ_j. Only the nodes for which there are moderate (neither too small nor too large) amount of node data are selected and their tied-shifts Δ_j are used for adaptation. This principle is depicted in Figure 6. When the total amount of data for adaptation is very small, the parameter Δ_j of the top node is applied to all the means in HMMs. This tie-shift represents a movement of the whole of the means in the acoustical feature space from the SI means. As the total amount of data increases, the nodes at the lower levels are selected for adaptation. The tied-shifts Δ_j of these nodes represent more local movements.

Some related studies were AMCC-MDL[5], in which MDL principle were used for AMCC, and Structural MAP (SMAP)[6], in which AMCC was combined with the adaptation approach based on MAP (Maximum A Posteriori) estimation.

4 MDL approach for Acoustic Modeling

In most speaker-independent speech recognition systems that employ continuous density HMMs, triphones are used as recognition units. While the large number of triphones can help to capture variations in speech data, the amount of available training data is likely to be insufficient to support the use of such a large number. This data insufficiency often causes serious degradation in speech recognition performance and most recognition systems using triphones cluster the model parameters to try to alleviate this problem. One of the most successful approaches is that based on the maximum-likelihood (ML) criterion. This ML approach, however, does not provide any solutions for determining the number of clusters (i.e., the optimal model size). To optimize it, most systems employ a series of recognition experiments, which is computationally expensive. The MDL approach[7], which uses the minimum description length(MDL) criterion in stead of ML criterion, was proposed to solve this problem. When a set of models $\{1, \ldots, i, \ldots, I\}$ is given, the description length $l_i(x^N)$ for data $\{x^N = x_1, \ldots, x_N\}$ and an underlying model i is given by

$$l_i(x^N) = -\log P_{\hat{\theta}^{(i)}}(x^N) + \frac{\alpha_i}{2} \log N + \log I \tag{1.3}$$

where α_i is the dimensionality (the number of free parameters) of model i and $\hat{\theta}^{(i)}$ represents the maximum likelihood estimates for the parameters $\theta^{(i)} = (\theta_1^{(i)}, \ldots, \theta_{\alpha_i}^{(i)})$ of model i. The first term on the right-hand side of (1.3) is the negative of the log-likelihood for data x^N when model i is used, and the second term is related to the size of model i and the number of data samples, N. As a model becomes larger, the value of the first term decreases and that of the second term increases. The second term works as a penalty imposed for employing a large model size. The MDL criterion selects the model with the minimum description length as the optimal model.

In the MDL approach for acoustic modeling, we use state splitting based on phonetic decision trees as a clustering scheme, . Those states at the same position in triphone HMMs having the same central phone are pooled into one set, and one

phonetic decision tree is constructed for each set. Starting from the root node which represents the whole set, each node from top to bottom splits off into two other nodes representing, respectively, "yes" or "no" answers to such phonetic-context related questions as: "Is the previous phone unvoiced?" (L-unvoiced?) and "Is the next phone a fricative?" (R-fricative?). The MDL criterion is used to choose the optimal question to be asked at each node and to decide when to stop splitting. When the root node S_0, is split into M nodes, $S_1, \ldots, S_M$, one model $U(S_1, \ldots, S_M)$ is defined for the node set $\{S_1, \ldots, S_M\}$. Using several approximations, the description length $l(U)$ for this model(node set) is calculated as follows:

$$l(U) = \sum_{m=1}^{M} \frac{1}{2}\Gamma_m(K + K\log(2\pi) + \log|\mathbf{\Sigma}_m|) + KM\log\sum_{m=1}^{M}\Gamma_m + C. \qquad (1.4)$$

where, K is the dimensionality of the input data, Γ_m is the state occupancy count at node S_m over all the data samples, Σ_m is the covariance of the Gaussian distribution at node S_m The second term corresponds to the threshold for the likelihood increase in ML approach, and is estimated automatically on the basis of the training data. The description length for each node set is calculated and the node set with the minimum description length is selected from among various node sets as being the optimal one.

5 Robust Recognition and Environment Adaptation

In this section, we describe NEC's original environment adaptation technique *rapid environment adaptation algorithm based on spectrum equalization* (REALISE)[8]. In practical speech recognition applications, differences between training and testing environments often seriously diminish recognition accuracy. These environmental differences can be classified into two types: difference in additive noise and difference in multiplicative noise in the spectral domain.

Fig. 7 shows an adaptation/recognition scheme using REALISE. The preliminary recognizer selects the closest reference pattern to the testing utterance from all reference patterns, and calculates time-alignment between the testing utterance and the closest reference pattern. The environmental difference estimator calculates environmental differences between the input and the reference pattern according to the time-alignment. When the environmental differences are extracted, all reference patterns are adapted to the input environment and the testing utterance is recognized again using the adapted reference patterns. This scheme premises that the input speech is one of the target vocabularies for speech recognition.

We assume there are two types of environmental noise sources which degrade speech recognition performance: additive noise and multiplicative noise. Additive noise is caused by various user environments (e. g. machinery noises, speech from others, etc.), and multiplicative noise is caused by filtering processes (e. g. microphones, transmission channels, the vocal tracts of individual speakers, etc.).

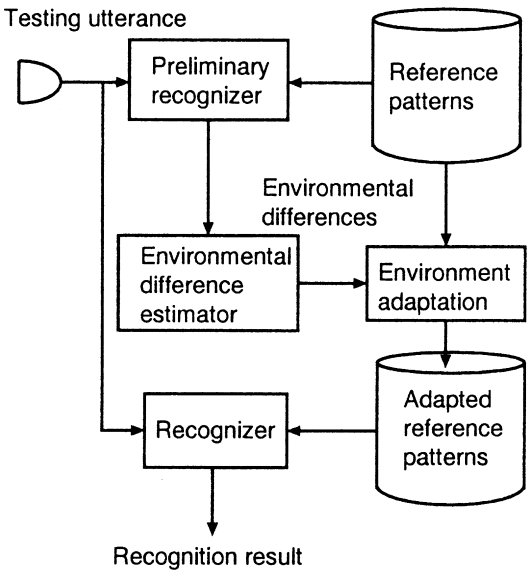

Figure 7: Speech adaptation/recognition scheme using REALISE

We introduce models in which both an input speech and a reference pattern are distorted by their own additive noise $\mathbf{B}$ and multiplicative noise $\mathbf{A}$. Assuming that $\mathbf{A}$ and $\mathbf{B}$ are constant within an utterance, we have

$$\begin{cases} \mathbf{V}(k) = \mathbf{A}_v \tilde{\mathbf{V}}(k) + \mathbf{B}_v \\ \mathbf{W}(k) = \mathbf{A}_w \tilde{\mathbf{W}}(k) + \mathbf{B}_w, \end{cases} \tag{1.5}$$

where k indicates the frame number, $\mathbf{V}(k)$, $\mathbf{W}(k)$, $\tilde{\mathbf{V}}(k)$, and $\tilde{\mathbf{W}}(k)$ are the observed spectra for the input and the reference pattern, and the undistorted spectra for the input and the reference pattern, respectively. Suffixes v and w indicate the input and the reference, respectively. Multiplicative noises $\mathbf{A}_v$ and $\mathbf{A}_w$ are diagonal matrices.

The goal of REALISE is to estimate spectra which are newly distorted by the input environment. From Eq. (1.5), we formulate the distorted spectrum $\hat{\mathbf{W}}(k)$ as follows:

$$\begin{aligned} \hat{\mathbf{W}}(k) &= \mathbf{A}_v \tilde{\mathbf{W}}(k) + \mathbf{B}_v \\ &= \mathbf{A}_v \mathbf{A}_w^{-1} (\mathbf{W}(k) - \mathbf{B}_w) + \mathbf{B}_v. \end{aligned} \tag{1.6}$$

By using the time-alignment, two additive noises, $\mathbf{B}_v$ and $\mathbf{B}_w$, can be calculated directly from the average of noise portion of the input and the reference pattern. On the other hand, $\mathbf{A}_v$ and $\mathbf{A}_w$ cannot be calculated directly, but can be related to the averages of speech portions.

Finally, from Eqs. (1.6), we obtain the goal using observable average spectra as

$$\hat{w}^i(k) \simeq \frac{s_v^i - n_v^i}{s_w^i - n_w^i} (w^i(k) - n_w^i) + n_v^i \tag{1.7}$$

where $w^i(k)$ and $\hat{w}^i(k)$ are element-wise representations for $\mathbf{W}(k)$ and $\hat{\mathbf{W}}(k)$ respectively, n_v^i and n_w^i are average noise portion of the input and the reference pattern respectively, and s_v^i and s_w^i are average speech portion of the input and the reference pattern respectively.

6 Speech recognition systems

We have developed a variety of speech recognition systems where demi-syllable unit based recognition was used. The main features of the systems are speaker-independent or speaker-adaptive, and are flexible as regards vocabulary (i.e. user utterances are not required).

6.1 Speech recognition system for LSIs

Speech recognition systems for LSIs which are suitable for use in electronics equipment such as car navigation systems, cellular phones or home computer game machines. The systems are speaker-independent and flexible as regards vocabulary, thus enabling users to define recognition vocabulary merely by depicting pronunciations using kana-character strings. In addition, the systems have an environmental adaptation function which minimizes the influence of various environmental noises(e.g noise generated by automobiles).

There are two types of systems. One is a system for DSP chips. This system recognizes a 100-word vocabulary in real-time. One example application for this system is name dialing (also called "voice search") in cellular phones. In this application, a user can call persons they wish to contact merely by speaking their names, which are registered beforehand, into the set.

The other type is a large vocabulary speech recognition system for a high-speed RISC processor, which is to be used for various kinds of multimedia applications. The system recognizes a more-than-100K-word vocabulary, e.g., as all city/town names in Japan, in real-time. The large vocabulary function is especially well suited for entering destinations in car navigation systems. In addition to car navigation, the system is expected to be applied to various kinds of user-friendly equipment such as mobile information terminals, PDAs or set top boxes.

6.2 Speech recognition server for telecommunication applications

Computer telephony integration is an area where speech recognition is particularly useful. A large-vocabulary speaker-independent telephone speech recognition server recognizes speech via cellular phone systems as well as conventional public telephone lines. Its 5,000-word vocabulary can be expanded up to 200K words by switching vocabulary dictionaries.

To achieve user-friendly interface in telephone application, a barge-in function is introduced, by which users can interrupt the audio response of the system. This is

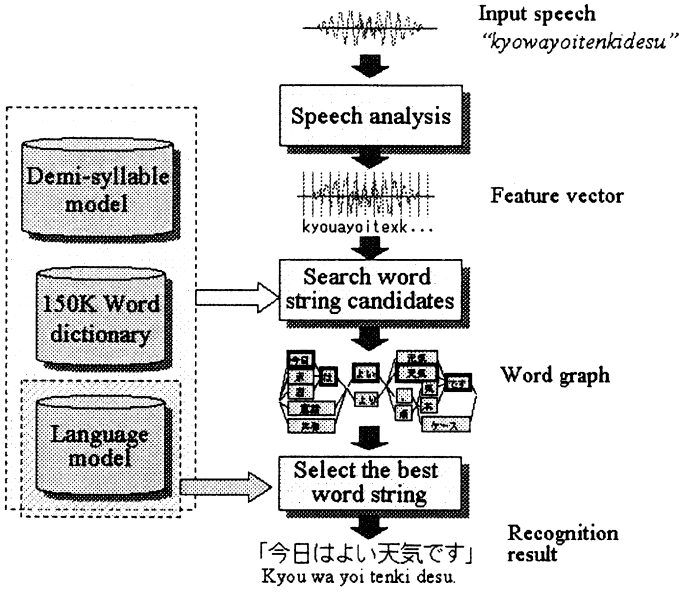

Figure 8: Dictation processing.

done by canceling an echo signal of the system's audio signal in the speech. A word spotting function is also introduced to allow users to speak as freely as possible.

Recognition servers are used in flight information services, sports/concert information services by telephone and automatic operator services. Such servers can also be used as compact automatic attendant systems for small offices.

6.3 Speech recognition software for PCs

Speech recognition software for personal computers is incorporated into various PC applications. The software has two main functions, "voice navigation" and "dictation".

Voice navigation enables users to control applications or to input string/numeric data into application software by voice. Speaker-independent discrete and continuous word recognition is performed, controlled by finite-state grammars defined by the applications.

Dictation allows the users to enter Japanese kanji-kana text by voice. The recognition software has a 150K-word dictionary that includes place names, landmarks, personal names and company names. Speaker adaptation is relatively easily done, through the utterance of only five sentences.

Figure 8 shows the block diagram of the dictation process.

7 Summary

NEC Corporation has developed a speaker-independent continuous speech recognition system that uses a demi-syllable speech unit and is applicable to a large vocabulary. To enable high performance speech recognition, high-speed computation methods are used for search processing in the continuous speech recognition and likelihood calculation. Speaker adaptation and environmental adaptation techniques enable robust speech recognition. The methods are used in speech recognition systems for LSIs, telecommunication systems and PCs.

Bibliography

1) T. Watanabe et. al., Speaker-independent speech recognition based on Hidden Marokov model using demi-syllable units, Systems and Computers in Japan, Vol.24, No.13, pp. 43-54 (1993)

2) T. Watanabe et. al., High-speed continuous speech recognition using a bundle search algorithm, Systems and Computers in Japan, Vol.24, No.13, pp. 65-75 (1993)

3) T. Watanabe et. al., High speed speech recognition using tree-structured probability density function, In *Proc. ICASSP95*, pp. 556–559 (1995)

4) T. Watanabe et. al., Unknown Utterance Rejection Using Likelihood Normalization Based on Syllable Recognition, Systems and Computers in Japan, Vol.24, No.14, pp. 74–84 (1993)

5) K.Shinoda, et. al., "Speaker Adaptation with Autonomous Model Complexity Control by MDL principle," In *Proc. ICASSP96*, pp.717-720, 1996.

6) K. Shinoda, et. al., "Unsupervised adaptation using structural Bayes approach," In *Proc. ICASSP98*, pp. 793-796(1998).

7) K.Shinoda, et. al., "Acoustic Modeling Based on the MDL Principle for Speech Recognition" , In *Proc. EuroSpeech97*, pp. 99-102(1997)

8) K. Takagi, et al., Speech Recognition with Rapid Environment Adaptation by Spectrum Equalization, Proc. of ICSLP, Vol. 3, S18.10, pp. 1023-1026 (1994)

The Authors

Takao Watanabe, He joined NEC Corp. in 1974. He now is working on speech and language research at C&C Res. Labs. NEC Corp.

Kaichiro Hatazaki, He joined NEC Corp. in 1980. He now is working on speech and language research at C&C Res. Labs. NEC Corp.

Ken-ichi Iso He joined NEC Corp. in 1986. He now is working on speech and language research at C&C Res. Labs. NEC Corp.

Ryosuke Isotani He joined NEC Corp. in 1987. He now is working on speech and language research at C&C Res. Labs. NEC Corp.

Koichi Shinoda He joined NEC Corp. in 1989. He now is working on speech and language research at C&C Res. Labs. NEC Corp.

Keizaburo Takagi He joined NEC Corp. in 1989. He now is working on speech and language research at C&C Res. Labs. NEC Corp.