Connectionist Approaches to Large Vocabulary Continuous Speech Recognition

Hidefumi Sawai[1], Yasuhiro Minami[2], Masanori Miyatake[3],
Alex Waibel[4] and Kiyohiro Shikano[5]

[1] ATR Interpreting Telephony Research Laboratories
[2] Faculty of Science and Technology, Keio University
[3] Information and Communication Systems Research Center, Sanyo Electric Company
[4] School of Computer Science, Carnegie Mellon University
[5] Human Interface Laboratories, NTT Company

Summary: This paper describes recent progress in a connectionist large-vocabulary continuous speech recognition system integrating speech recognition and language processing. The speech recognition part consists of Large Phonemic Time-Delay Neural Networks (TDNNs) which can automatically spot all 24 Japanese phonemes (i.e., 18 consonants /b/, /d/, /g/, /p/, /t/, /k/, /m/, /n/, /N/, /s/, /sh/, /h/, /z/, /ch/, /ts/, /r/, /w/, /y/ and 5 vowels /a/, /i/, /u/, /e/, /o/ and a double consonant /Q/ or silence) by simply scanning among input speech without any specific segmentation techniques. Training the Large TDNN is performed based on a fast back-propagation procedure using shifted training tokens extracted from training word speech. On the other hand, the language processing part is made up of a predictive LR parser in which the LR parser is guided by the LR parsing table automatically generated from context-free grammar rules, and proceeds left-to-right without backtracking. Time alignment between the predicted phonemes and a sequence of the TDNN phoneme outputs is carried out by the DTW matching method. We call this 'hybrid' integrated recognition system the 'TDNN-LR' method. We report that large-vocabulary isolated word and continuous speech recognition using the TDNN-LR method provided excellent speaker-dependent recognition performance. Furthermore, we discuss the following currently developing issues: robustness for variations of *speaking manner*, *speaker-independent* phoneme recognition and *speaker-adaptation* problems as extensions of the TDNN-LR speech recognition system.

1. Introduction

In this paper, we describe recent progress in the connectionist large-vocabulary continuous speech reconition system we have developed at ATR. First, we review our research achievements: phoneme recognition, phoneme/syllable spotting techniques using Time-Delay Neural Networks(TDNNs). Second, we describe large-vocabulary and continuous speech recognition using TDNN phoneme spotting and LR parsing technique. Finally, currently developing techniques as extensions to speaker-independent phoneme recognition and speaker-adaptation techniques are described.

We have demonstrated that a TDNN performed excellent phoneme recognition for a small but difficult task, i.e., bdg-phoneme recognition. Scalling up such a subnetwork to a larger network is another critical problem. Scaling up connectionist models to larger connectionist systems is difficult, because large networks require increasing amounts of training time and data, and the complexity of the optimization task quickly reaches computationally unmanageable proportions. We trained several small TDNNs aimed at all phonemic subcategories (nasals, fricatives, vowel, etc.) and reported excellent fine phonemic subcategory networks. We then proposed several techniques that allow us to "grow" larger nets in an incremental and modular fashion without loss in recognition performance and without the need for excessive training time or additional data. These techniques include class discriminatory learning, connectionist glue, selective / partial learning and all-net fine tuning.

In section 3, we describe syllable and phoneme spotting experiments. Syllable or phoneme spotting if reliably achieved, provides a good solution to the spoken word and/or continuous speech recognition problem. We would like to extend the encouraging performance of TDNN to word/continuous speech recognition. We showed techniques for spotting Japanese CV syllables/phonemes in input speech based on TDNNs. We constructed a TDNN which could discriminate a single CV-syllable or phoneme. In Japanese, there are only about one hundred syllables, or less than thirty phonemes, which makes it feasible to prepare and train the TDNN to spot all possible syllables or phonemes. These spotting techniques proved to be a good step toward continuous speech recognition.

We have found that a phoneme spotting approach is more effective for recognizing continuous speech than a

CV-syllable spotting approach because there are fewer phonemes than CV-syllables, and because preparing phoneme training tokens is easier than preparing CV-syllable tokens. Also, a phoneme-based recognition method is better suited for a large-vocabulary system than word-template-based recognition methods. Since TDNNs have superior phoneme recognition performance and time-shift invariance, an accurate and efficient speech understanding system could be accomplished by adapting the TDNN spotting method to continuous speech recognition.

In section 4, we describe the integration of speech processing and language processing. The speech recognition part consists of the Large Phonemic TDNN which can automatically spot all 24 Japanese phonemes (i.e., 18 consonants /b/, /d/, /g/, /p/, /t/, /k/, /m/, /n/, /N/, /s/, /sh/, /h/, /z/, /ch/, /ts/, /r/, /w/, /y/ and 5 vowels /a/, /i/, /u/, /e/, /o/ and a double consonant /Q/ or silence) by simply scanning among an input speech without any specific segmentation techniques. The Large Phonemic TDNN architecture is constructed as 4-layered back-propagation type networks in a modular fashion, where a group of easily confused phonemes are integrated into sub-networks, and each sub-network is also integrated into one hidden layer. Training the Large TDNN is performed based on a fast back-propagation procedure[8] using shifted training tokens extracted from training-word speech and/or training continuous speech, because the shift-invariance property of the large TDNN was found to be effective in the region of 20-30 ms in a preliminary experiment[12].

On the other hand, the language processing part is made up of a predictive LR parser[17] in which the LR parser is guided by the LR parsing table automatically generated from context-free grammar rules, and proceeds left-to-right without backtracking. The predicted LR parser predicts subsequent phonemes based on the currently processed phonemes which are produced from the output units of the Large Phonemic TDNN scanning input speech. Time alignment between the predicted phonemes and a sequence of the TDNN phoneme outputs is carried out by a DTW matching method. A duration control technique is applied for the predicted phonemes during the DTW matching to appropriately constrain the alignment.

We call this "hybrid" integrated recognition system the "TDNN-LR" method. The TDNN-LR recognition system provides *vocabulary-independent*, large-vocabulary and continuous speech recognition because the Large Phonemic TDNN is trained by phoneme tokens extracted from various contexts of training word and/or continuous speech.

In section 5, two kinds of recognition experiments i.e., large-vocabulary isolated word recognition and continuous speech recognition, were performed using the TDNN-LR method. In section 6, we describe currently developing issues as extensions including robustness for variations of *speaking manner*, *speaker-independent* phoneme recognition and *speaker-adaptation* problems. We will introduce recent results surrounding those issues.

2. Phoneme Recognition Using TDNN

2.1. TDNN Architecture

For the recognition of phonemes, a three-layer net is constructed. Its overall architecture and a typical set of activities in the units are shown in Fig.1 based on one of the phonemic subcategory tasks (BDG).

At the lowest level, 16 melscale spectral coefficients serve as input to the network. Input speech, sampled at 12 kHz, was hamming windowed and a 256-point FFT computed every 5 msec. Melscale coefficients were computed from the power spectrum[1] and coefficients adjacent in time collapsed resulting in an overall 10-msec frame rate. The coefficients of an input token (in this case 15 frames of speech centered around the hand labeled vowel onset) were then normalized to lie between -1.0 and +1.0 with the average at 0.0. Fig.1 shows the resulting coefficients for the speech token "BA" as input to the network, where positive values are shown as black squares and negative values as grey squares. The detailed architecture is described in ref.[1].

We have used a large-vocabulary database of 5,240 common Japanese words[19]. The entire database was phonetically hand labeled. These labels were used in the experiments reported below and applied to learning and evaluation. The data used was uttered in isolation by one male native Japanese speaker. The database was then split into a training set and a testing set of 2,620 utterances each, from which the actual phonetic tokens were extracted. The training tokens (up to 600 tokens per phoneme) were randomized within each phoneme class. Training the TDNN was performed using a back-propagation learning procedure[7]. For performance evaluation, we have run all experiments on the testing tokens only, i.e., on tokens not included during training. The resulting data included a considerable amount of variability due to its position within an utterance or phonetic context. The TDNN achieved a recognition rate of 98.5% averaged for three male speakers[1].

2.2. Phoneme Recognition by Modular TDNN Design

Our consonant TDNN (shown in Fig.2) was constructed modularly from networks aimed at the consonant subcategories, i.e., the bdg-, ptk-, mnN-, sshhz-, chts- and the rwy-tasks. Each of these nets had been trained before to discriminate between the consonants within each class. In addition, an interclass discrimination net that distinguishes between the consonant subclasses was trained. This hopefully provides missing feature information for interclass discrimination. Three connections were then established to each of the 18 consonant output categories (/b/, /d/, /g/, /p/, /t/, /k/, /m/, /n/, /N/, /s/, /sh/, /h/, /z/, /ch/, /ts/, /r/, /w/ and /y/): one to connect an output unit with the appropriate interclass discrimination unit in hidden layer 2, one with the appropriate intraclass discrimination unit from hidden layer 2 of the corresponding subcategory net and one with the always-activated threshold unit (not shown in Fig.2). The overall network architecture is illustrated in Fig.2 for the case of an incoming test token (e.g., a /g/). For simplicity, Fig.2 shows only the hidden layers from the bdg-, ptk-, sshhz- and the interclass discrimination nets. At the output, only the two connections leading to the correctly activated /g/-output unit are shown.

All free weights were initialized with small random weights and then trained by the back-propagation learning procedure[7][8]. 96.7% of all tokens were correctly categorized into one of the six consonant subclasses. After completion of the learning run the entire net was evaluated over 3,061 consonant test

tokens, and achieved a 95.0% recognition accuracy. All-net fine tuning was then performed by freeing up all connections in the network to allow all connections to make small additional adjustments in the interest of better overall performance. After completion of the all-net fine tuning, the performance of the network then yielded 96.0% correct consonant recognition over the test data. Furthermore, a fast back-propagation method later developed at ATR made it possible to train the consonant network from random weight values at the same time, and yielded a better recognition rate of 96.7%[8].

3. Phoneme Spotting Using TDNN

3.1. Initial Atempts

Spotting CV-syllables is a good approach to word and continuous speech recognition in Japanese because there are only about one hundred syllables. The architecture of the TDNN is extremely suitable for spotting CV-syllables/phonemes because the shift-invariant structure makes it possible to correctly spot them even in the neighboring positions of syllable/phoneme tokens. If we train a neural network which can reliably discriminate one syllable, and also prepare all kinds of neural networks, spotting of any syllables, in principle, can be achieved for any input utterances. However, because there are about one hundred syllables which can be chosen, one significant problem will occur when we choose as training tokens all possible syllables except the specific syllable to be discriminated. As an initial attempt in spotting Japanese CV-syllables, we arbitrarily chose the syllable "BA" as a typical Japanese syllable. As training tokens, the "non-BA" syllables "DA", "GA", "PA", "TA" and "KA" are chosen because they might be confused with "BA". We could automatically discriminate all possible syllables other than the five syllables used as training tokens. As a result of that, both training time and the number of tokens could be greatly reduced. The spoting experiments on "BA" syllables showed that a rate of 96.7% was achieved, and other possible syllables except "BA" (not only "DA", "GA", "PA", "TA" and "KA" but also all other syllables) are well inhibited at a rate of 99.3%[2, 3].

In general, spotting phonemes is also an effective approach to speech recognition in other languages. We study the phoneme spotting approach for comparison with the syllable spotting approach. In this case, phoneme group networks which can discriminate one phoneme group (ex.:"BDG") are trained. "BDG", "PTK", "MNsN", "SShHZ", "ChTs", "RWY" and "AIUEO" are chosen as phoneme groups. As well as training phoneme group networks, the corresponding "intra-group" networks are also provided with training tokens of each subcategory (ex.:"B", "D" and "G" for the "BDG" intra-group network). Spotting experiments are performed by determining the phoneme category which is involved in the phoneme group. This approach to spotting phonemes is critical if an incorrect phoneme group were determined. However, it is advantageous that only 7 phoneme groups and 7 intra-group networks need be prepared rather than preparing about one hundred CV-syllable networks. Even though some misfired patterns appeared in phoneme spotting because the network were trained only with easily confused training tokens extracted from the center positions of phonemic tokens, excellent spotting results were obtained[2].

These results in spotting Japanese-CV syllables and phonemes in words strongly suggested that these spotting techniques can be applied to recognizing not only spoken words but also continuous speech.

3.2. A Large TDNN Architecture for Spotting Phonemes

A large TDNN architecture for discriminating 24 Japanese phonemes (18 consonants: /b/, /d/, /g/, /p/, /t/, /k/, /m/, /n/, /N/, /s/, /sh/, /h/, /z/, /ch/, /ts/, /r/, /w/, /y/, and 5 vowels /a/,/i/, /u/, /e/, /o/, and silence) was constructed as shown in Fig.3[3, 6, 11, 12]. This TDNN is modulay constructed by 6 intra-class subnetworks discriminating among "bdg", "ptk", "mnN", "sshhz", "chts" and "rwy", an intra-class subnetwork discriminating among consonant groups, a vowel network, and a silence network discriminating between silence and speech. The input layer consists of 240 units, i.e., 16 melscale filterbank coefficients * 15 frames (10 ms frame rate) as same as the bdg-net. These subnetworks are integrated into a third hidden layer which has 24 units so that their corresponding output units can be laterally inhibited.

3.3. Training the Large TDNN

Phoneme tokens for training the TDNN are classified into 24 phoneme categories, based on hand labeling, extracted from even-numberd words of the 5,240 common words uttered by the same male speaker as in section 2. Phoneme training tokens are extracted from various positions of each phoneme in order to avoid inclination of phoneme positions i.e., a duration of 150 ms from each phoneme is extracted by shifting its center by 20 ms, while phoneme boundary tokens are eliminated. Tokens for silence are extracted from the points 20 ms before the beginning or after the end of words. The number of training tokens per phoneme category ranges up to 1,000, randomly selected from the extracted tokens. Tokens are duplicated when the number per category can not reach 1,000. Each token is analyzed under the same conditions as in section 2.

Training the TDNN is performed using a fast back-propagation learning procedure[8]. This algorithm is based on a steep error surface, updating weight parameters frequently, and omitting computation for samples with small error. Using this learning procedure made it possible to simultaneously train the network in Fig.3 without modular training.

3.4. Phoneme Spotting Experiments

Phoneme spotting outputs are obtained as recognition results by shifting the input layer among input speech frame by frame. This phoneme spotting method does not require any phoneme segmentation techniques and can get spotting results merely by scanning the network. [2, 3, 11, 12]

Applying the trained TDNN to the 2,620 test words, it is found that most phonemes are correctly spotted throughout each phoneme segment, and that the outputs corresponding to the other categories are well inhibited. The lower layer shows an input spectrogram and the upper shows spotting outputs, where the horizontal axis is time and the vertical axes in the lower and the upper layers represent frequency and type of phoneme, respectively.

Table 1 shows spotting resuts when using up to 400 and 1,000 training tokens/category, respectively. It is demonstrated that 98.0% of the phonemes in the test words are correctly spotted for the latter case, yielding a false alarm rate of 23.2%.

Thus, the Large Phonemic TDNN is already trained by as many as 18,864 training tokens extracted from 2,620 training words. For the first experiment, spotting experiments in continuous speech were conducted using the TDNN. The initial correct phoneme spotting rate in 278 Japanese test phrases was 81.2% with a false alarm rate of 47.8%, as shown in Table 2. Because of the different co-articulatory effects of word speech and continuous speech, incremental TDNN training using a small number of tokens extracted from continuous training speech seemed to be needed. The number of tokens for incremental training is only 100/200 tokens per phoneme category (2,011/3,251 tokens are only 11%/17% of the original tokens extracted from the training words). The correct phoneme spotting rate was significantly improved from 81.2% to 89.0%/89.1% after the adaptive incremental training. More importantly, the false alarm rate decreased from 47.8% to 34.8%/25.8%. Fig. 4 shows an example of phoneme spotting results in the phrase /touroku-wo/. We can also expect better phrase recognition rates in continuous speech after the incremental training.

4. The TDNN-LR Recognition System

To extend the high performance spotting results to large-vocabulary continuous speech recognition, a "hybrid" method combining a predictive LR parser[17] with a DTW alignment technique was proposed. We applied this method to 5,240 common Japanese words and phrases[19] uttered by the male speaker.

4.1. LR Parser

Though there are other possible methods which can match TDNN spotting results with reference patterns, we used a predictive LR parsing (as an extension of a generalized LR parsing[16]) method because it is applicable to sentence recognition. The LR parsing method is available for ambiguous sentences which could not be dealt with by an ordinary LR parsing method[12].

LR parsing is well known in the field of program languages, and is applicable to a large class of context-free grammars. Generalized LR parsing[16] is a kind of LR parsing, and has been extended to handle arbitrary context-free grammars. For an ambiguous grammar, the LR parsing table has multiple entries. The LR parser is guided by the LR parsing table automatically created from context-free grammar rules, and proceeds left-to-right without backtracking. These parsing algorithms are very efficient for natural language processing.

The LR parsing method includes FIFO(First In First Out) which keeps track of the stack status, saving the states of the LR parser, an ACTION table and a GOTO table. These tables are generated by a context-free grammar. The LR parser processes an input string by referencing these tables and the stacks. The GOTO table determines an action by using an input string and its own status. The actions are: "shift", "reduce", "accept" and "error";

shift : insert the state of the parser into the top of the stack

reduce : sumarize stack status

accept : complete analysis

error : failure of analysis

The following are analysis procedures ;

[Definition]

s : a status of the parser

a, A : grammar symbols(non-terminal, terminal symbols)

input pointer : a currently processed input string

status stack : preserve the status of the parser

GOTO(s,a) : return the value of the next status by referencing the status "s" and the gramar symbol "a" in the GOTO table

ACTION(s,a) : determine an action by referencing the status "s" and the gramar symbol "a" in the ACTION table

<Algorithm>

(1) Intialization

Set the input pointer to the top of an input string and then push "0" in a status stack.

(2) Check the ACTION(s, a) by referncing the current status "s" and the input pointer symbol "a".

(3) If ACTION(s,a) = "shift", push GOTO(s,a) into the status stack and then advance the input pointer by one.

(4) If ACTION(s,a) = "reduce, n", pop up the stack status on the right side of the n-th grammar rule. If the top of the stack status is "s'", the next status GOTO(s',A) is pushed to the stack by referencing s' and a grammar rule "A" on the left side of the n-th grammar rule. These "pop up" and "push down" actions are procedures summarizing the right side to the left side in the grammar rules.

(5) If ACTION(s,a) = "accept", the analysis is completed.

(6) If ACTION(s,a) = "error", the analysis failed.

(7) Return to (2).

An example of sentences in the LR parsing and a LR parsing table are shown in Fig. 5 and Fig.6, respectively[17]. The LR table consists of an ACTION table and a GOTO table. In Fig. 6, lines show grammar symbols, and columns show parser status. The symbols "s" and "r" show "shift" and "reduce" actions, respectively. The figure on the right side of "s" is the next status in a "shift" action. The figure on the right side of "r" is the number of grammar rules. The right side shows a GOTO table where the figure indicates the next status value.

A generalized LR parser can analyze ambiguous sentences which could not be dealt with by an ordinary LR parser. Though actions in the LR parser describe a single action in an element of the table, the generalized LR parser can describe multiple actions in one element. Processing multiple actions in parallel makes it possible to deal with ambiguity in a grammar[16].

4.2. A Predictive LR Parser

A predictive LR parsing method predicts the next phonemes in input speech based on the currently processed phonemes. An HMM continuous speech recognition system using an predictive LR parsing has been evaluated[7]. This technique is also applicable to spotting results from the TDNN and a word or phrase grammar describing a large vocabulary or phrase database[19], respectively.

A predictive LR parser analyzes a sentence by predicting subsequent phonemes. Prediction can be easily realized by referencing an LR table such as Fig.5. By way of analysis, when the predictive LR parser is in a status, possible phonemes served to this LR parser are the only phonemes described by "shift" and "reduce" on a line of the table. The predictive LR parser regards these phonemes as predicted phonemes.

4.3. Integration of the TDNN and the Parser

The basic structure of the recognition system which utilizes TDNN spotting and predictive LR parsing is shown in Fig.6 (hereafter: TDNN-LR). First, an input speech is converted to outputs via TDNN phoneme spotting shown in the upper part of Fig.4. Matching between these outputs and reference words is performed by the predictive LR parser according to a grammar rule. The grammar rule is registered as an LR table in advance based on a context-free grammar. The predicted LR parser predicts the next phonemes based on already processed phonemes. When plural phonemes are predicted, the predictive LR parser analyzes the phonemes in parallel. The predicted phoneme sequences are evaluated by a DP match between predicted phonemes and the TDNN phoneme spotting results. This procedure continues until input phonemes come to an end. However, since it takes considerable time to process all predicted phonemes, a beam search is used to take the first "B" candidates, where "B" is the width of the beam.

The likelihood of a similarity between a predicted phoneme and an input phoneme is defined as the logarithm of the activation value of TDNN output. The length of the reference patterns (predicted phoneme patterns) is the average length of the training phoneme tokens extracted from the training words of the large vocabulary. The slope constraint in DTW alignment is 1/2 to 2. The matching algorithm is as follows:

j : predicted phoneme
p(t, j): output value of phoneme "j" at frame "t"
D0(t): table#0 for saving likelihood
D1(t): table#1 for saving likelihood

[Initialization]

$Q(0, t) = D0(t)$, $Q(1, t) = D1(t)$, $t = 1,2,...,N$.
$Q(1, 1) = p(1, j)$, otherwise 0.

(Iterative formula)

for $t = 1,...,N$ and $i = 1,...,M$.

$$Q(i, t) = \max [\, Q(i-1, t-1) + \log(p(t, j)),$$
$$Q(i-2, t-1) + \log(p(t, j)) + \log(p(t, j)),$$
$$Q(i-1, i-2) + 0.5 * \log(p(t-1, j)) + 0.5 * \log(p(t, j))]$$
$$D0(t) = Q(M-1, t), \quad D1(t) = Q(M, t).$$

The above process is a DP match of predicted phonemes with their end points free. After this procedure, subsequent phonemes are predicted with initial values of D0(t) and D1(t), and then the next DP match is performed. When plural phonemes are predicted, parallel processing is conducted. Phoneme sequences are built up step by step using this DP matching procedure, and recognition results are obtained by taking the maximum value of candidates within the beam width. This procedure is similar to a level-building DTW match[18] with its endpoints free, which builds up subsequent phonemes.

5. Recognition Experiments

5.1. Large Vocabulary Speech Recognition

In recognition experiments of large vocabulary, 5,240 common Japanese words were used. Among those words, another half of the large database which were not used for the network training were used as test words. The number of test words was incremented as 100, 500, 2,620 test words. On the other hand, the number of reference words was also incremented as 100, 500, 2,620 and 5,240 words, where in the former three cases, the reference words corresponded to the test words, and in the last case, the 5,240 reference words included the 2,620 test words as a subset. Therefore, note that this experiment is vocabulary-independent recognition.

Fig.7 shows the recognition rates of the n-th ($1 \leqq n \leqq 5$) top choices as a function of the vocabulary size of reference words from 100 to 5,240. In the case of the whole 5,240 words, a rate of 92.6% is obtained for the top choices, and rate of 97.6% and 99.1% are obtained for the second and fifth choices, respectively. While the rate of top choices decreases according to the increased vocabulary size, the rate within the top 5 choices is maintained higher than 99.1% for any vocabulary size.

Recognition error in 5,240 common words is classified into the following three cases: (1) insertion of "t" or "k" at the beginning of a word (ex.: "aisuru" → "taisuru"). (2) a short word is misrecognized as a long word (ex.:"aa" → "hanahada"). (3) a double consonant is confused with a silence accompanied by an unvoiced stop(affricate) (ex.:"itai" → "ittai").

Case (1) occurred due to the fact that TDNN easily inserted "t" or "k" at the unstable beginning of a word. Case (2) is due to the fact that a matching path is beyond the limitation of DP matching. Case (3) is due to the fact that the difference in durations between a double consonant and a silence accompanied by an unvoiced stop(affricate) is not introduced into the DP matching method.

5.2. Continuous Speech Recognition

The Large Phonemic TDNN is already trained by as many as 18,864 training tokens extracted from 2,620 training words. As an first attempt, continuous speech recognition experiments were conducted using the trained TDNN and an LR-parser describing *general* phrase grammar rules (its phoneme-perplexity is 5.9). Table 3 shows the features of the ATR "Conference Registration" task we used. The initial phrase recognition rate for 278 Japanese test phrases was 55.0% for the top choices and 82.7% for the top 5 choices, respectively. Because of different co-articulatory effects between word speech and continuous speech, incremental training of the TDNN using a small number of training tokens extracted from continuous training speech seemed to be needed.

The number of training tokens for incremental training is only 100 tokens per phoneme category (2,011 tokens in total are only 11% of the original tokens extracted from the training words). We then increased the number up to 200 tokens per category (3,251 tokens in total). The phrase recognition rates are shown in Table 4 as compared with the rates before the incremental training. A phrase recognition rate of 65.1 % for the top choices and 88.8% for the top 5 choices

were obtained. Therefore, the efficiency of adaptive incremental training using a small number of training tokens extracted from continuous speech was confirmed through this experiment.

Typical errors are as follows:

(1) Substitution errors between /n/ and /m/.

e.x. : /saNka-no/ → /saNka-mo/,
/syotei-no/ → /syotei-mo/.

These errors occurred due to the fact that the number of /m/ and /n/ phoneme tokens for incremental training was too small (17 tokens for /m/ and 13 tokens for /n/) compared with the original training tokens extracted from traininig words (1,000 for /m/ and 460 for /n/).

(2) Phoneme insertion errors.

e.x. : /zyuusho/ → /zyuusho-o/,
/happyou/ → /happyou-o/.

These errors occurred due to difficulty of precise duration control at the end of the utterances.

6. Extensions

In this section, we describe extensions in the TDNN-LR speech recognition system; *robustness for variations of speaking manner, speaker-adaptation and speaker-independent phoneme recognition.*

6.1. Neural Network Architectures for Robust Speech Recognition

Until now, Time-Delay Neural Networks (TDNN) architecture has been applied to several speaker-dependent recognition stages, such as phoneme recognition(described in section 2), Japanese CV-syllable/phoneme spotting (in section 3), and the TDNN-LR large-vocabulary continuous speech recognition system with integrated training for spotting Japanese phonemes (in section 4). If we extend these recognition methods based on TDNN to a continuous, *speaker-independent* speech recognition system, a novel robust recognition strategy should be developed. This section introduces several novel TDNN architectures for robust *speaker-independent*, continuous speech recognition[20, 21].

One novel architecture for a Frequency-shift-invariant TDNN (FTDNN) is based on the frequency-time-shift-invariance as well as the time-shift-invariance by constructing the same weighting values between the input layer and the hidden layers of the TDNN. Speech features from the input layer of the FTDNN are individually extracted along the time-axis and the mel-scaled frequency-axis by each corresponding first hidden layer. The extracted features are then integrated into a single second hidden layer. The final decision is made based on the activation patterns whose property is invariant from both the time- and frequency-shift of input phoneme tokens.

Another novel architecture is a Block-Windowed NN (BWNN), based on windowing each layer of the NN with local time-frequency windows. This architecture makes it possible for the NN to capture global features from the upper layers as well as precise local features from the lower layers, because the local windows in the upper layers can integrate more global features than those in the lower layers. A five layered BWNN is constructed for a phoneme recognition experiment.

Very confusable phoneme recognition experiments were performed using /b/, /d/, /g/, /m/, /n/, and /N/ (syllabic nasal) phoneme tokens to verify robustness toward variations of speech. The FTDNN and BWNN are trained by phoneme tokens extracted from word speech (utterance speed : 5.65 mora/s), and tested using unknown test phoneme tokens extracted from test word speech (same utterance speed as the training tokens) and continuous speech (utterance speed : 9.56 mora/s). Performance among an original TDNN, a FTDNN and a BWNN was compared using the same training and test phoneme tokens. Recognition rates of 96.7% for the FTDNN and 98.2% for the BWNN were obtained compared with a rate of 95.9% for the original TDNN, and rates of 80.8% and 82.8% for the FTDNN and BWNN, respectively, which are significantly better than a rate of 68.1% for the original TDNN.

6.2. Speaker-adaptation Using Neural Networks

Speaker-adaptation is one good approach to a *speaker-independent* recognition problem. It is neccessary to use a small amount of training data uttered by an input speaker to adapt a speech recognition system. A speaker-adaptation technique using neural networks have been proposed[22]. It is also possible to use segmental speech for speaker-adaptation by building a mapping function from an input speaker to a standard speaker. We proposed a segmental approach using neural network identity mapping as a supervised learning method[23]. In this approach, segmental speech including a phoneme or syllable can be mapped between two speakers through a neural network and DTW matching method[22, 23]. This mapping network can be used as a front end of the TDNN-LR speech recognition system.

As a preliminary experiment using the speaker-adaptation neural network and a TDNN for recognizing voiced stops i.e., /b, d, g/[23], we performed speaker-adaptation experiments between two male speakers. Before speaker-adaptation, a recognition rate was 86.2%, which was improved to 91.0% after speaker-adaptation. This technique is being applied to other phoneme categories including all consonants and phonemes. Also, an unsupervised speaker-adptation technique using neural networks is being investigated[24].

6.3. Speaker-independent Recognition

In this section, we compare several TDNN architectures applied to *speaker-dependent* and *multi-speaker's* phoneme recognition with respect to their capabilities in a *speaker-independent* recognition problem.

We verified performance of several architectures: (1)single TDNN, (2)SID(Stimulus Identification) network, (3)Meta-Pi network, (4)Modular TDNN and (5)Modular Speaker ID network, where the single TDNN is an original architecture, the SID network is constructed by both each speaker's module and a speaker ID module which selects outputs in each speaker's module, the Meta-Pi network is reported as the network most suitable for *multi-speaker* phoneme recognition[25]. However, it has not been demonstrated how the Meta-Pi network is effective for a *speaker-independent* phoneme recognition problem. Furthermore, two novel modular TDNN architectures ((4) & (5)) are proposed to improve the performance. The modular TDNN is a network which is constructed by integrating each speaker's module (i.e., a single TDNN) trained on the first stage, and retrained on the second stage to recognize each phoneme, regardless of training speakers. The Modular Speaker ID network comprises

of a speaker ID module in addition to the Modular TDNN, thus explicitly classifying each speaker ID as in the Meta-Pi network.

Speaker-independent phoneme experiments for recognizing voiced stops /b, d , g/ using six and twelve training speakers showed high recognition rates of 92.1% for the modular TDNN and 95.6%, respectively for the Modular Speaker ID network. These results are significantly better than the rates of 82.0% and 85.9%, respectively for the Meta-Pi network. As a result, it is found that the Meta-Pi architecture suitable for *multi-speaker* recognition is not necessarily robust for a *speaker-independent* recognition task. The recognition rate for the Modular Speaker ID network nearly matches the *speaker-dependent* recognition rate of 98.0% for the single TDNN[26, 27].

7. Conclusion

We described an integration of speech recognition and language processing. The speech recognition part consists of the Large Phonemic Time-Delay Neural Networks (TDNN) which can automatically spot all 24 Japanese phonemes with an excellent spotting rate of 98.0% by simply scanning among an input speech along with it. The language processing part is made up of a predictive LR parser which predicts subsequent phonemes based on the currently processed phonemes. The TDNN-LR hybrid recognition system provides large-vocabulary and continuous speech recognition. Two kinds of recognition experiments i.e., large-vocabulary isolated word recognition and continuous speech recognition were performed using the TDNN-LR method. Speaker-dependent recognition rates of 92.6% for the first choices and 97.6% for the top two choices were obtained for 5,240 Japanese common words, and rates of 65.1% for the first choices and 88.8% within the fifth choices were attained for phrase recognition.

We also described several neural network approaches for *robust speech recognition, speaker-adaptaion* and *speaker-independent* speech recognition as extensions of the TDNN-LR speech recognition system.

Acknowledgement

The authors would like to express their gratitude to Dr. Akira Kurematsu, president of ATR Interpreting Telephony Research Laboratories, for his support, and to Dr. Shigeki Sagayama, Head of Speech Processing Department for his valuable discussions on this research. They are also indebted to the members of the Speech Processing Department at ATR, for their constant help in various stages of this research.

References

[1] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano and K.Lang,"Phoneme Recognition Using Time-Delay Neural Networks," IEEE Trans. on ASSP, Vol.37, No.3, pp328-339, Mar. 1989.

[2] H. Sawai, A. Waibel, M. Miyatake and K. Shikano,"Spotting Japanese CV-Syllables and Phonemes Using Time-Delay Neural Networks," IEEE, Proceedings of ICASSP-89, S1.7, May 1989

[3] H. Sawai, A Waibel, P. Haffner, M. Miyatake and K. Shikano,"Parallelism, Hierarchy, Scaling in Time-Delay Neural Networks for Spotting Japanese Phonemes/CV-Syllables," Int. Joint Conf. on Neural Networks, Proceedings of IJCNN-89, vol. II , pp81-88, June 1989..

[4] A. Waibel, H. Sawai and K. Shikano,"Consonant Recognition by Modular Construction of Large Phonemic Time-Delay Neural Networks," IEEE, Proceedings of ICASSP-89, S3.9, May 1989.

[5] A. Waibel, H. Sawai and K. Shikano, "Modularity and Scaling in Large Phonemic Neural Networks" IEEE Trans. on ASSP, Vol. 37, No.12, pp1888-1898, Dec. 1989.

[6] H. Sawai, A. Waibel, M. Miyatake and K. Shikano, "Phoneme Recognition by Scaling up Modular Time-Delay Neural Networks," IEICE Technical Report SP88-105, 1988.

[7]D. E. Rumelhart, J. E. McClelland and the PDP Research Group, "Parallel Distributed Processing," MIT Press (1986).

[8] P. Haffner, A. Waibel, H. Sawai and K. Shikano,"Fast Back-Propagation Learning Methods for Large Phonemic Neural Networks," European Conference on Speech Communication and Technology, pp553-556, Paris, Sep. 1989.

[9] Y. Minami, M. Miyatake, H. Sawai and K. Shikano, "Continuous Speech Recognition Using TDNN Phoneme Spotting and Generalized LR Parser," Proceedings of ASJ Fall Meeting, 3-1-11, 1989,

[10] Y. Minami, H. Sawai and M. Miyatake, "Large Vocabulary Spoken Word Recognition Using Time-Delay Neural Network Phoneme Spotting and Predictive LR Parsing," J. of IEICE, vol. J73-D-II, no.6, pp788-795, Jun, 1990.

[11] M. Miyatake, H. Sawai, Y. Minami and K Shikano, "Integrated Training for Spotting Japanese Phonemes Using Large Phonemic Time-Delay Neural Networks," IEEE, Proceedings of ICASSP-90, S8.10, Apr. 1990.

[12] M. Miyatake, H. Sawai and K. Shikano, "Training Methods and Their Effects for Spotting Japanese Phonemes Using Time-Delay Neural networks," J. of IEICE, vol. J73-D-II, no.5, pp699-706, May, 1990.

[13]H. Sawai, "Effect of Incremental Training in the TDNN-LR Phrase Speech Recognition System," Proceedings of ASJ Fall Meeting, 2-P-11, Sep. 1990.

[14]H. Sawai, "TDNN-LR Large-Vocabulary and Continuous Speech Recognition System," Proceedings of ICSLP-90, vol.2, pp1349-1352, Nov. 1990.

[15]H. Sawai, "TDNN-LR Continuous Speech Recognition System Using Adaptive Incremental TDNN Training," Proceedings of ICASSP-91, May. 1991, to be presented.

[16]M . Tomita,"Efficient Parsing for Natural Language - A Fast Algorithm for Practical Systems," Kluwer Academic Publishers (1986).

[17] K. Kita, T. Kawabata and H. Saito,"HMM Continuous Speech Recognition Using Predictive LR Parsing," IEEE, Proceedings of ICASSP-89, S13.3, May 1989.

[18] C. S. Myers and R. Labiner:"A Level Building Dynamic Time Warping Algorithm for Connected Word Recognition," IEEE Trans. on ASSP, vol.29, No.2, pp284-279 1981.

[19] H. Kuwabara, K. Takeda, Y. Sagisaka, S. Katagiri, S. Morikawa and T. Watanabe, "Construction of a Large-Scale Japanese Database and Its Management System," IEEE, Proceedings of ICASSP-89, S10b.12, May 1989.

[20]H. Sawai, "Frequency-Time-Shift-Invariant Time-Delay Neural Networks for Robust Continuous Speech Recognition," Proceedings of ICASSP-91, May. 1991, to be presented.

[21]H. Sawai, "Time-Frequency Shift Tolerant Time-Delay Neural Networks," Proceedings of ASJ Fall Meeting, 2-P-2, Sep. 1990.

[22]K. Iso, M. Asogawa, K. Yoshida and T. Watanabe, "Speaker Adaptation Using Neural Network," Proceedings of ASJ Spring Meeting, 1-6-16, Mar. 1989.

[23]K. Fukuzawa, H. Sawai and M. Sugiyama, "Speaker Adaptation Using Identity Mapping by Neural Networks," Proceedings of ASJ Fall Meeting, 1-8-16, Sep. 1990.

[24]M. Sugiyama, K. Fukuzawa, H. Sawai and S. Sagayama, "Unsupervized Training Methods for Set Mappings Using Neural Networks," Proceedings of ASJ Fall Meeting, 2-P-10, Sep. 1990.

[25]J. Hampshire, A. Waibel, "The Meta-Pi Network : Connectionist Rapid Adaptation for High-Performance Multi-Speaker Phoneme Recognition", Proceedings of the 1990 IEEE International Conference on Acoustics, Speech and Signal Processing, S3.9, pp164-168 1990.

[26] S. Nakamura, H. Sawai, "Speaker-Independent Phoneme Recognition Using Time-Delay Neural Networks," ATR Technical Report, TR-I-0178, Sep. 1990.

[27] S. Nakamura, H. Sawai, "A Preliminary Study on Neural Network Architectures for Speaker-Independent Phoneme Recognition," IEICE Technical Report, SP90-61, Dec. 1990.

[30] A. Waibel, "Connectionist Large Vocabulary Speech Recognition," J. of IEICE, vol. J73-D-II, no.8, pp1122-1131, Aug. 1990.

Figure Captions:

Fig.1 The TDNN Architecture (input: "BA")

Fig.2 Modular Construction of all Consonant Network

Fig.3 The Large Phonemic TDNN Architecture

Fig.4 An Example of Spotting Results: (phrase name is /touroku-wo/)

Fig.5 An Example of Context-Free Grammar

Fig.6 An Example of ACTION and GOTO Tables

Fig.7 The TDNN-LR Speech Recognition System

Fig.8 Results on Large-Vocabulary Recognition

Table Captions:

Table 1 TDNN Phoneme Spotting Results on Large-Vocabulary

Table 2 TDNN Phoneme Spotting Results on Test Phrases

Table3 Features of the Task

Table4 Phrase recognition rates(%)

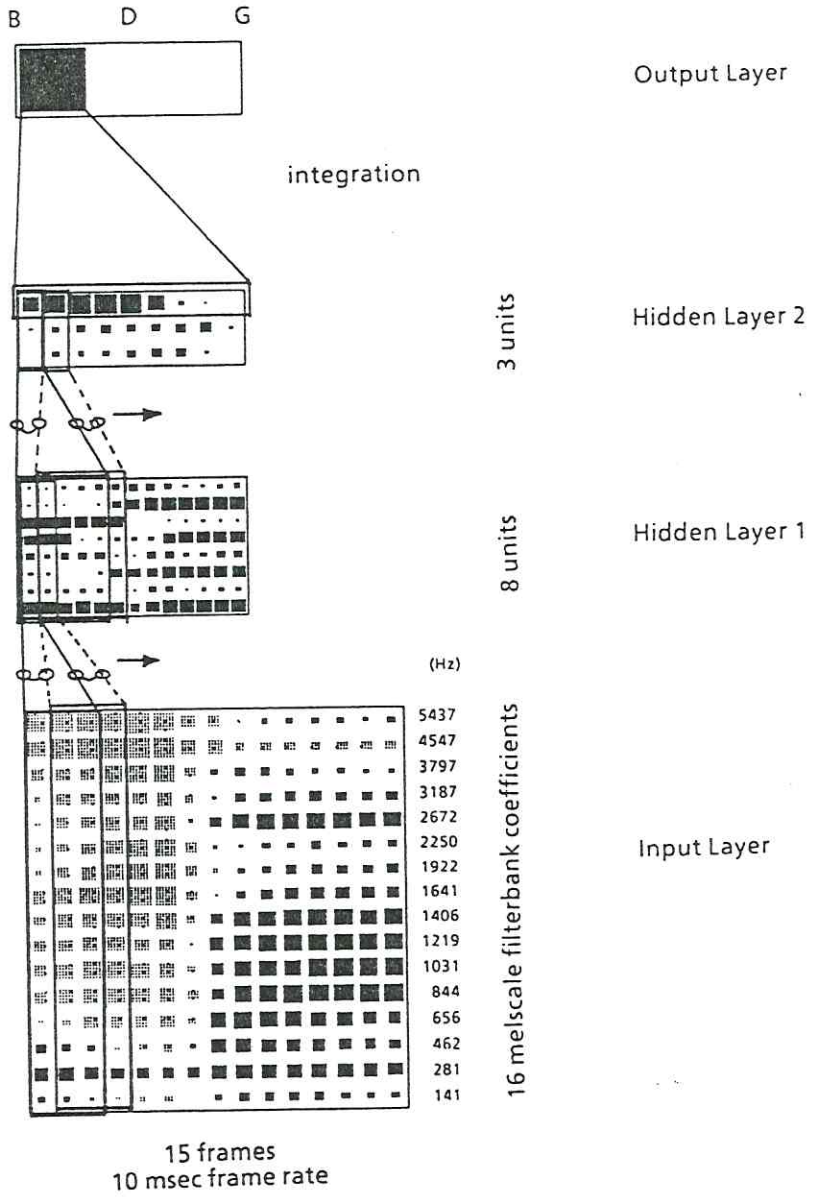

Fig.1 The TDNN Architecture (input: "BA")

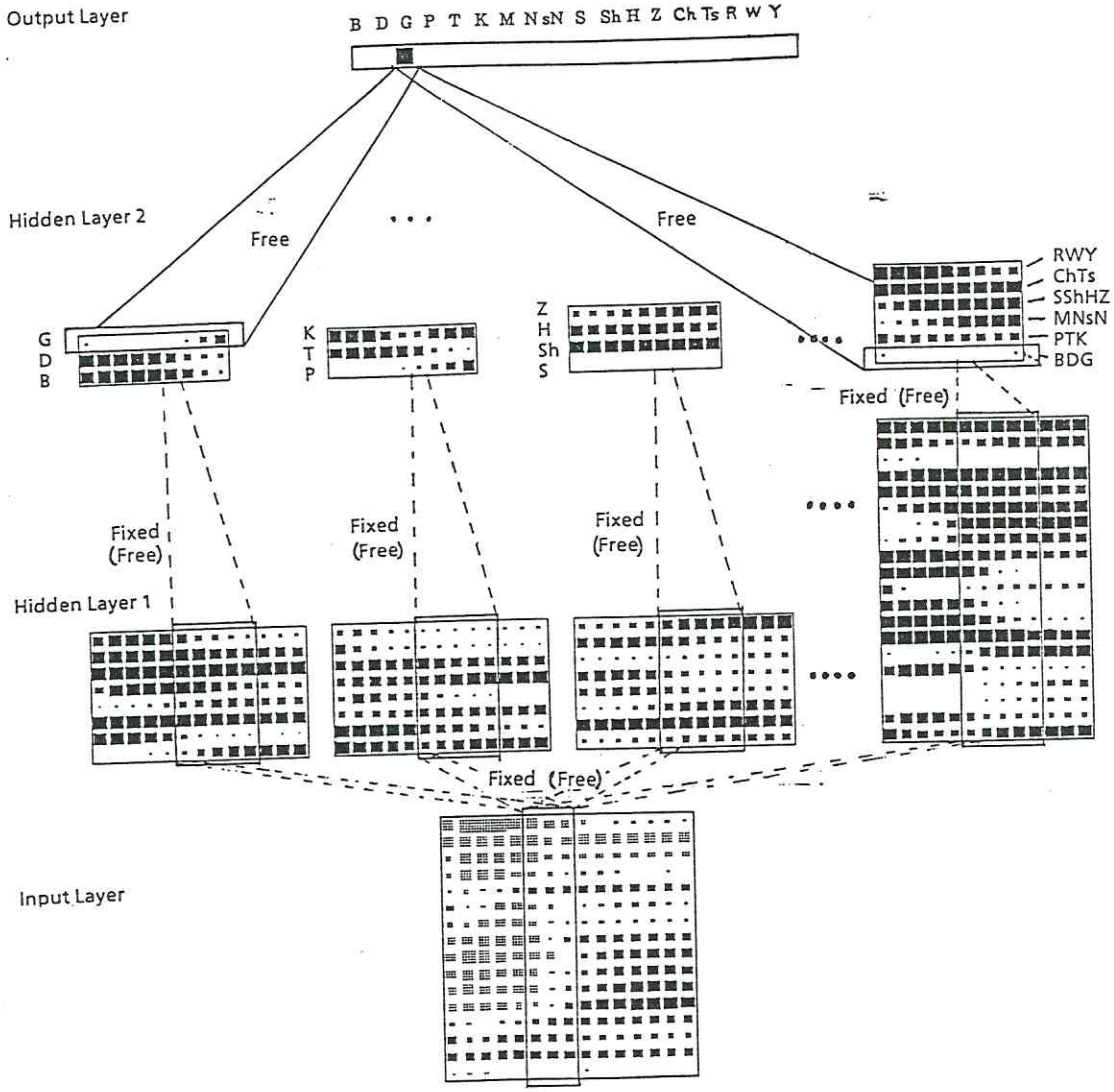

Fig.2 Modular Construction of all Consonant Network

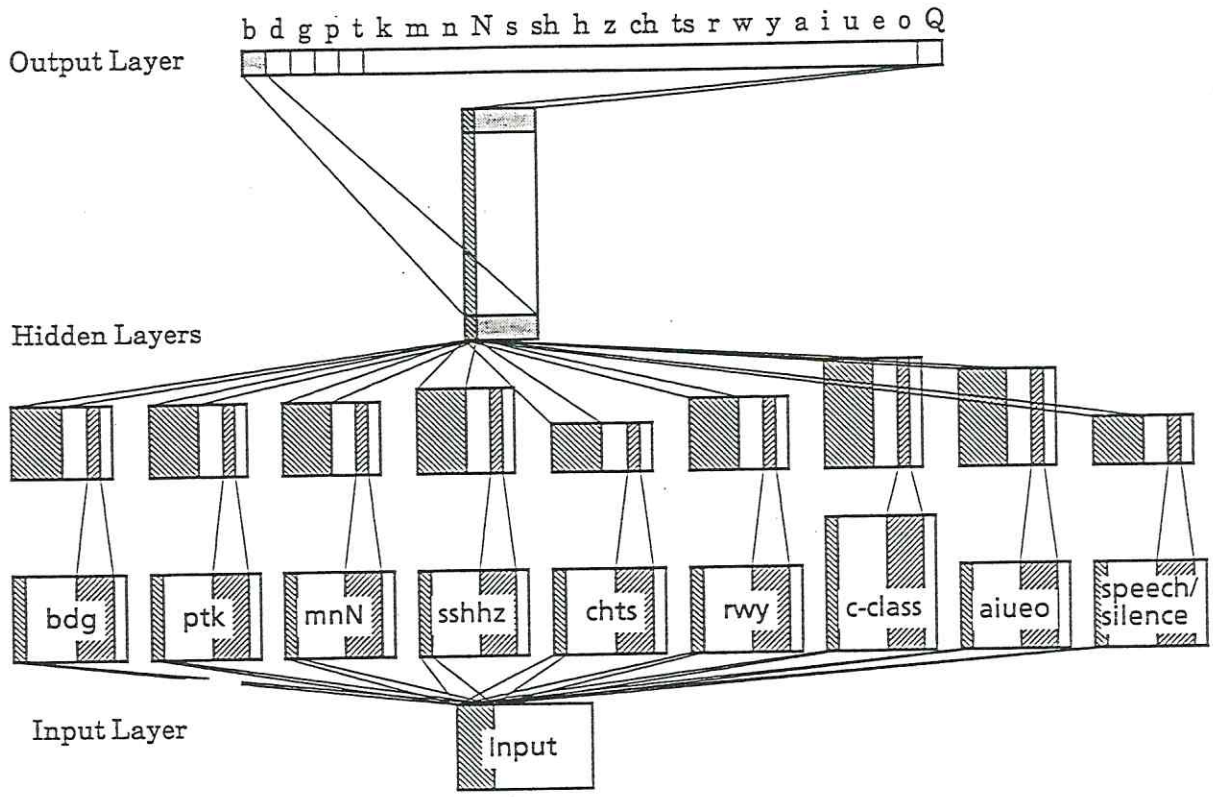

Fig.3 The Large Phonemic TDNN Architecture

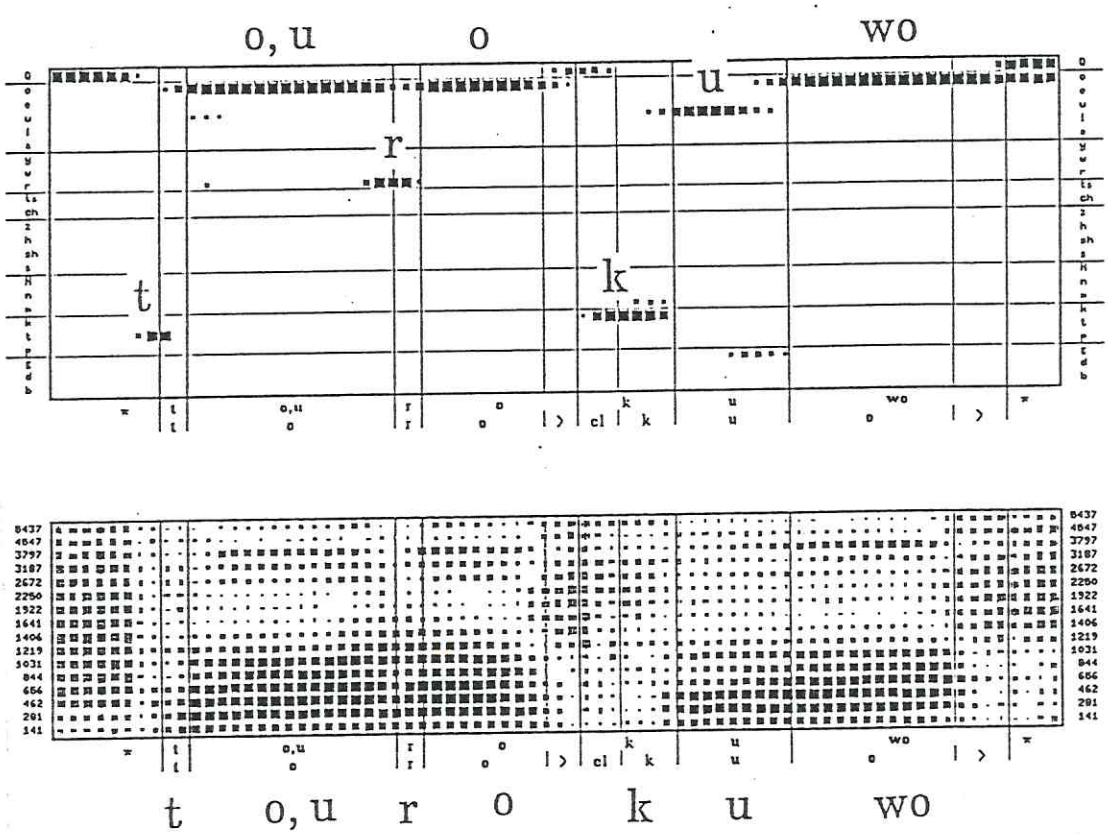

Fig.4 An example of phoneme spotting results:
(phrase name is /touroku-wo/.)

(1) S	→	NP V
(2) NP	→	N
(3) NP	→	N P
(4) N	→	m a m e
(5) N	→	a r e
(6) P	→	o
(7) V	→	o k u r e
(8) V	→	k u r e

Fig.5 An Example of Context-Free Grammar

	a	u	e	o	k	m	r	$	S	N	V	P	NP
0	s2					s3			5	4			1
1				s7	s6						8		
2							s9						
3	s10												
4				s11,r2								12	
5								acc					
6		s13											
7					s14								
8								r1					
9			s15										
10						s16							
11				r6									
12				r3									
13							s17						
14		s18											
15				r5									
16			s19										
17			s20										
18							s21						
19				r4									
20								r8					
21			s22										
22								r7					

Fig.6 An Example of ACTION and GOTO Tables

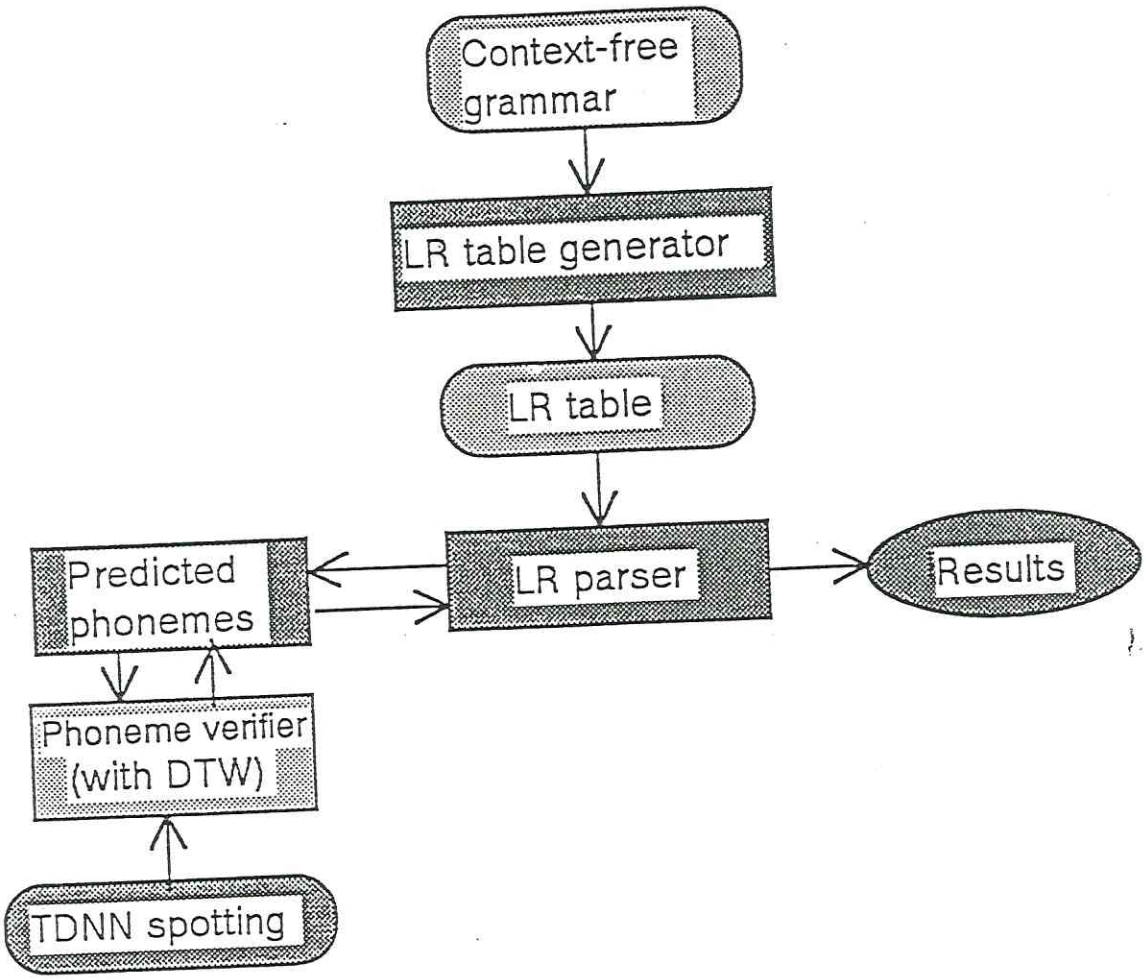

Fig.7 The TDNN-LR Speech Recogntion System

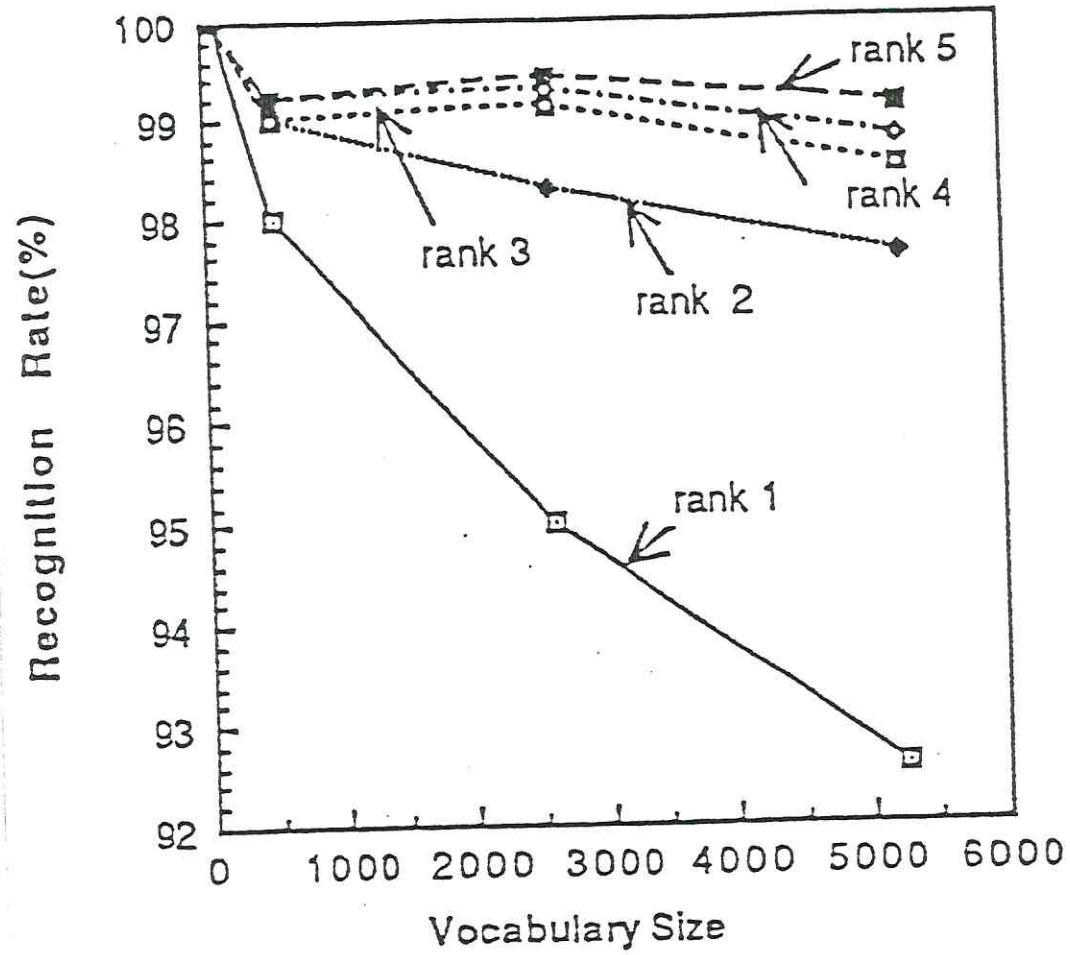

Fig.8 Results on Large-Vocabulary Recognition

Table 1 TDNN Phoneme Spotting Results on Large-Vocabulary

Phon-emes	#of phonemes	400tokens/category			1,000 tokens/category		
		Correct	Deletion	Insertion	Correct	Deletion	Insertion
b	231	228	3	268	225	6	104
d	180	175	5	106	171	9	71
g	265	230	35	198	210	55	57
p	28	25	3	203	26	2	104
t	461	452	9	178	459	2	235
k	1300	1218	82	116	1283	17	245
m	485	482	3	323	479	6	213
n	273	258	15	84	263	10	63
N	488	487	1	161	488	0	163
s	572	570	2	175	572	0	100
sh	387	385	2	52	386	1	81
h	313	312	1	215	310	3	159
z	315	310	5	170	307	8	87
ch	141	140	1	57	141	0	163
ts	220	219	1	205	218	2	235
r	760	709	51	62	730	30	97
w	81	80	1	74	79	2	13
y	573	531	42	124	561	12	171
a	1772	1770	2	108	1771	1	85
i	1333	1282	51	155	1302	31	200
u	1615	1496	119	206	1543	72	200
e	829	822	7	222	827	2	254
o	1352	1337	15	97	1348	4	136
Total	13974	13518 (96.7%)	456 (3.3%)	3559 (25.5%)	13699 (98.0%)	275 (2.0%)	3236 (23.2%)

Table 2 TDNN Phoneme Spotting Results on Test Phrases

Phon-eme	#of phonemes	No adaptive training			Adaptive training (200 tokens/cat.)		
		Correct	Deletion	insertion	Correct	Deletion	insertion
b	16	14	2	19	10	6	5
d	69	44	25	29	62	7	19
g	34	19	15	36	19	15	17
p	10	8	2	6	10	0	6
t	70	48	22	11	68	2	56
k	210	182	28	94	195	15	7
m	58	36	22	19	18	40	23
n	74	36	38	5	33	41	11
N	34	24	10	25	27	7	19
s	74	67	7	20	72	2	9
sh	53	50	3	15	53	0	17
h	27	14	13	10	24	3	48
z	32	32	0	53	32	0	25
h	17	10	7	11	16	1	7
ts	24	24	0	79	22	2	13
r	66	53	13	43	61	5	24
w	25	5	20	3	23	2	14
y	96	80	16	30	90	6	30
a	279	238	41	23	272	7	16
i	192	160	32	27	179	13	39
u	97	90	7	287	85	12	50
e	127	107	20	38	121	6	27
o	256	234	22	43	237	19	18
total	1940	1575 (81.2%)	365 (18.8%)	926 (47.8%)	1729 (89.1%)	211 (10.9%)	500 (25.8%)

Table3 Features of the Task

Number of words	1,035
Number of rules	1,656
Number of states in LR	5,015
Phoneme perplexity	5.9
Entropy/ phoneme	2.6 bit
Average number of phonemes/ phrase	7.32

Table4 Phrase recognition rates(%)

Rank	Before adaptive training (without duration control)	Before adaptive training (with duration control)	After adaptive training (100/cat)	After adaptive training (200/cat)
1	52.9	55.0	64.4	65.1
2	70.1	70.1	79.5	78.4
3	77.7	76.6	81.7	87.1
4	81.7	81.3	86.0	88.1
5	82.4	82.7	88.8	88.8
6~10	86.3	87.1	93.2	91.0
11~15	87.4	87.4	93.5	92.4
16~	12.6	12.6	6.5	7.6

Connectionist Approaches to Large-Vocabulary Continuous Speech Recognition

Hidefumi SAWAI[1], Yasuhiro Minami[2], Masanori Miyatake[3], Alex Waibel[4] and Kiyohiro Shikano[5]

1 ATR Interpreting Telephony Research Laboratories
2 Faculty of Science and Technology, Keio University
3 Information and Communication Systems Research Center, Sanyo Electric Company
4 School of Computer Science, Carnegie Mellon University
5 Human Interface Laboratories, NTT Company

This paper describes connectionist approaches to large-vocabulary continuous speech recognition integrating speech recognition and language processing. The speech recognition part consists of the Large Phonemic Time-Delay Neural Networks (TDNNs) which can automatically spot all Japanese phonemes by simply scanning among an input speech. The language processing part is made up of a predictive LR parser which predicts subsequent phonemes based on currently processed phonemes. Recognition experiments using ATR's large-vocabulary speech database with 5,240 words and "Conference Registration" task, yielded high recognition performance. Furthermore, we discuss some extensions of the current system for robust speech recognition, speaker-adaptation and speaker-independent recognition.