

論文 / 著書情報
Article / Book Information

Title	Continuous speech Recognition Using a Dependency Grammar and Phoneme-Based HMMs
Authors	Sho-ichi Matsunaga, Shigeru Homma, Shigeki Sagayama, Sadaoki Furui
出典 / Citation	Trans. IEICE, Vol. 74, No. 7, pp. 1826-1833
発行日 / Pub. date	1991,
URL	http://search.ieice.org/
権利情報 / Copyright	本著作物の著作権は電子情報通信学会に帰属します。 Copyright (c) 1991 Institute of Electronics, Information and Communication Engineers.

Continuous Speech Recognition Using a Dependency Grammar and Phoneme-Based HMMs

Sho-ichi MATSUNAGA[†], Shigeru HOMMA^{††}, Shigeki SAGAYAMA^{††}
and Sadaoki FURUI[†], *Members*

SUMMARY This paper describes two Japanese continuous speech recognition systems (system-1 and system-2) based on phoneme-based HMMs and a two-level grammar approach. Two grammars are an intra-phrase transition network grammar for phrase recognition, and an inter-phrase dependency grammar for sentence recognition. A joint score, combining acoustic likelihood and linguistic certainty factors derived from phoneme-based HMMs and dependency rules, is maximized to obtain the best sentence recognition results. System-1 is tuned for sentences uttered phrase-by-phrase and system-2 is tuned for sentence utterances, to make the amount of computation practical. In system-1, two efficient parsing algorithms are used for each grammar. They are a bi-directional network parser and a breadth-first dependency parser. With the phrase-network parser, input phrase utterances are parsed bi-directionally both left-to-right and right-to-left, and optimal Viterbi paths are found along which the accumulated phonetic likelihood is maximized. The dependency parser utilizes efficient breadth-first search and beam search algorithms. For system-2, we have extended the dependency analysis algorithm for sentence utterances, using a technique for detecting most-likely multi-phrase candidates based on the Viterbi phrase alignment. Where the perplexity of the phrase syntax is 40, system-1 and system-2 increase phrase recognition performance in the sentence by approximately 6% and 14%, showing the effectiveness of semantic dependency analysis.

1. Introduction

In Japanese sentences, the phrase order is much less constrained than in English. On the other hand, the word order within phrases (called "BUNSETU") is very regular, and the sentence structure is ordered by semantic dependency (called "KAKARIUKE") between phrases. Syntactic constraints are useful in recognizing specific tasks or short-duration utterances. However, particularly in sentence recognition for phrase-order-free languages such as Japanese, semantic constraints are more powerful than sentence syntactic constraints. Thus, it is considered that combining different types of these grammars is practical for linguistic processing of Japanese speech recognition. Recently, several dependency analysis algorithm have

been developed⁽¹⁾⁻⁽³⁾.

On the other hand, the hidden Markov model (HMM) was introduced, and achieved good results in speech recognition^{(4),(5)}. Our goal is to develop an effective method of combining high-level linguistic knowledge and refined phonetic models to improve speech recognition results. Therefore, we have developed two Japanese continuous speech recognition systems that obtain the most likely sentence results taking account of acoustic, syntactic, and semantic factors based on phoneme-based HMMs and a two-level grammar approach⁽⁶⁾. Each system uses two grammars: an intra-phrase transition network grammar for phrase recognition, and an inter-phrase dependency grammar for sentence recognition. The former is a syntactic grammar, and the latter is a semantic and loose syntactic grammar. The dependency grammar is compatible with the case grammar. Thus, semantic certainty factor is determined taking into account grammatical cases.

System-1 is tuned for sentences uttered phrase-by-phrase and system-2 for sentence utterances, to make the amount of computation practical. In system-1, two efficient parsing algorithms are used for each grammar. They are a bi-directional network parser and a breadth-first dependency parser. With the phrase-network parser, input phrase utterances are parsed bi-directionally both left-to-right and right-to-left, and optimal Viterbi paths are found along which the accumulated phonetic likelihood is maximized. The parser generates a phrase matrix. With the dependency parser, inter-phrase dependency structures within a sentence are analyzed. The joint score, obtained by combining accumulated phonetic likelihood and semantic certainty factor, is maximized to obtain the solution on this matrix. The dependency parser utilizes efficient breadth-first search and beam search algorithms.

For system-2, we have extended the dependency analysis algorithm for sentence utterances, using a technique for detecting most-likely multi-phrase candidates based on the Viterbi phrase alignment. Taking account of pauses between phrases, syntactic constraints of sentences are represented by multi-connection of these networks. The network parser

Manuscript received January 10, 1991.

Manuscript revised March 25, 1991.

[†] The authors are with NTT Human Interface Laboratories, Musashino-shi, 180 Japan.

^{††} The author is with NTT Tohoku Software Plan and Produce Center, Sendai-shi, 980 Japan.

^{†††} The author is with ATR Interpreting Telephony Research Laboratories, Kyoto-fu, 619-02 Japan.

frame-synchronously parses input sentence utterances to get a lattice of the multi-phrase candidates and their likelihoods for the selected phrase boundaries extracted by the Viterbi algorithm, and generates a phrase lattice. The dependency parser also analyzes inter-phrase dependency structures on the lattice.

The approach of these two systems is very suitable for speech understanding systems since it can use semantic dependency structures.

2. Two-Level Grammar Approach

2.1 Intra-Phrase Grammar

A Japanese phrase is composed of a stem part and a suffix part. The stem part is a verb, a noun, or an adjective. The suffix part is composed of suffixes such as auxiliary verbs and case markers called "JOSHI". The connection of these parts is so regular that phrase syntax is represented by networks. However, there are many kinds of connection rules. Therefore, to cover concisely this variety, the syntactic structure is represented by transition networks (TNs) composed of many sub-networks.

2.2 Dependency Structure Grammar

After the phrase likelihood has been calculated, recognition results are represented in a matrix form or in a lattice form. When analyzing dependency relationships, the parser extracts the most likely sentence by taking into account the phonetic likelihood of phrase candidates and the linguistic certainty factor of semantic inter-phrase dependency relationships. The parser also obtains the dependency structure that corresponds to the semantic structure of the extracted sentence.

Dependency grammar is based on semantic dependency relationships between phrases. The syntactic rules satisfy the following two constraints. First, every phrase, except the last phrase, must modify one and only one later phrase. This modification is called a dependency relationship or dependency structure. It is represented by an arc frequently. Second, modification arcs between phrases must not cross.

Because dependency structure grammar is compat-

ible with case grammar, the semantic certainty factor of the dependency structure is provided using case structure. This is described in detail in reference⁽³⁾. Certainty factor is composed of two categories. One is the certainty factor associated with dependency relationships of only the modifier and modificant phrases: disagreement between the semantic primitive of the modifier and that required by the modificant, disagreement between the case of modifier and that required by the modificant, incorrect idiomatic expression and so on. The other is the certainty factor associated with all the dependency structure of the phrase sequence: lack of a phrase of the obligatory case required by the modificant, modification of the same phrase by different phrases having the same case category, complexity of dependency structure and so on. The certainty factor values for these items are given as penalties (negative values) heuristically.

3. Speech Recognition System-1

A block diagram of system-1 is shown in Fig. 1. Input sentences are uttered phrase by phrase. After feature parameter extraction from the utterance, the parameter sequence is converted into a vector code sequence. Next, phonetic likelihood is calculated for every possible duration to obtain likelihood matrices for phoneme candidates, based on HMMs. Phonetic duration time can be easily controlled by giving maximum and minimum duration times to each phoneme using the matrix. Next, phrase likelihood is calculated based on the phonetic likelihood matrices. The verb and adjective entries in the dictionary have grammatical cases, and the noun entries are accompanied by semantic primitives. Next, the top candidates of each likelihood are generated in a matrix form. The number of candidates is optional. Finally, using this matrix and the dependency grammar, the parser extracts the most likely sentence of a phrase sequence and its dependency structure.

3.1 Speech Recognition of Phrases

If the phrase parser expands the transition networks to a single network in phrase recognition proc-

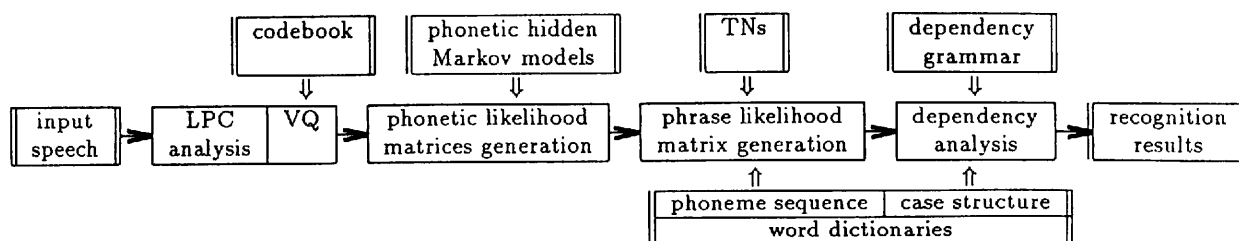


Fig. 1 Block diagram of the recognition system (system-1) for continuous speech uttered phrase by phrase.

ess, the network becomes too large to handle and too time consuming because of the complexity of phrase structure. To overcome these problems, a bi-directional parser was developed. Sagayama et al. developed the original bi-directional approach for connected word recognition using the DP algorithm⁽⁷⁾. Our parser parses the stem part left-to-right, and suffix part right-to-left. Bottom-up parsing is carried out for the stem and the last suffix of the suffix part, and top-down parsing is carried out for inner suffix called "KAT-SUYOO". By calculating the top-most m candidates for each sub-network, the top-most m phrase candidates are generated based on the Viterbi algorithm.

A simple example of phrase syntax and flow of parsing are shown in Fig. 2. S and B correspond to syntax networks, and A , C , D and E correspond to word dictionaries. Processing flow is $A \rightarrow D \rightarrow E \rightarrow C \rightarrow B \rightarrow S$, and its processing sequence is shown in the figure. First, the stem words, corresponding to A , are parsed left-to-right from the beginning of speech. Next, the last suffixes, corresponding to D and E , are processed right-to-left from the end of speech. The parsing result of B is generated by combining the results of D , E and C . The likelihood calculation of C , namely likelihoods of $c1$ and $c2$, is carried out at every top-down scan of C . Finally, the phrase recogni-

tion result is generated by combining the results of A and B . In the figure, $\rightarrow$, $\leftarrow$, $\uparrow$, $\downarrow$ indicate the left-to-right process, the right-to-left process, the bottom-up control, the top-down control, respectively.

3.2 Performance of a Bi-directional Parser

The performance of the bi-directional parser (BD method) is evaluated by comparing two-level dynamic programming approach using the Cocke-Kasami-Younger algorithm (TD method)^{(8),(9)} and frame synchronous parsing approach (FS method)^{(10),(11)}. The two levels of the TD method are calculation stages of phonetic likelihood and that of phrase likelihood. Each parser's performance is slightly different when each HMM is learnt using the Baum-Welch algorithm. The performance of TD using a forward trellis computation is the highest. That of BD using the forward computation for intra-phoneme likelihood derivation and the Viterbi algorithm for inter-phoneme is next. That of FD is third using the Viterbi algorithm only. Processing amount of the forward trellis is much larger than that of the Viterbi algorithm.

Comparison experiments are carried out using input phrases whose duration is about one second. Three kinds of linguistic constraints are used whose number of word entries are 360, 130 and 20. The top-five candidates are generated in each parsing. Processing time ratio is shown in Table 1. These results show that the BD method is more effective than the FS method when vocabulary size is larger. This is because the BD method looks up the phoneme likelihood matrices which are calculated in advance. Furthermore, the TD method needs much more computation than the other methods, and it is considered that the BD method is one of the most useful parsing methods using phoneme-based HMMs for Japanese phrase recognition owing to recognition performance and parsing amount.

3.3 Parser for Dependency Structure Grammar

Dependency parsing is equivalent to solving the following objective function using the constraints of dependency structure grammar.

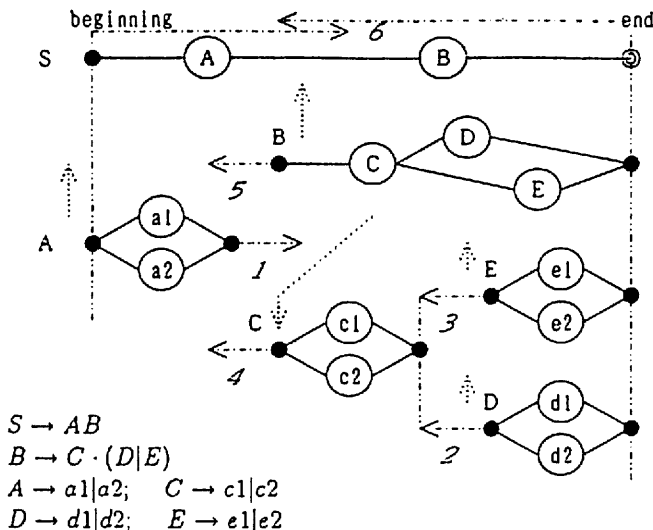


Fig. 2 Example of phrase recognition process.

Table 1 Processing amount ratio for phrase recognition.

parsing approach	BD	FS	TD
HMM recognition method (intra-phoneme)	trellis	Viterbi	trellis
(inter-phoneme)	Viterbi	Viterbi	Viterbi (Trellis is possible)
performance order	2	3	1
word entries (360 words)	16	18	626
word entries (130 words)	12	12	-
word entries (20 words)	5	1	-

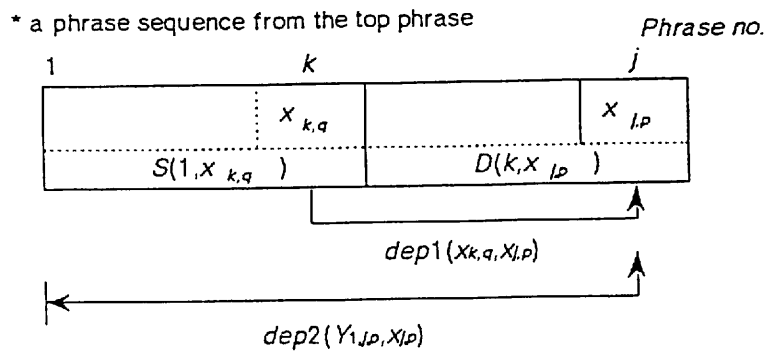
Processing amount is normalized by that of FS method using 20 words-dictionary.

$$T = \max_{\{p\}} \left[\sum_{j=1}^N c(x_{j,p}) + \max_Y \sum_{j=1}^N \text{dep}(w_{1,j-1}, x_{j,p} | Y_{1,j,p}) \right] \quad (1)$$

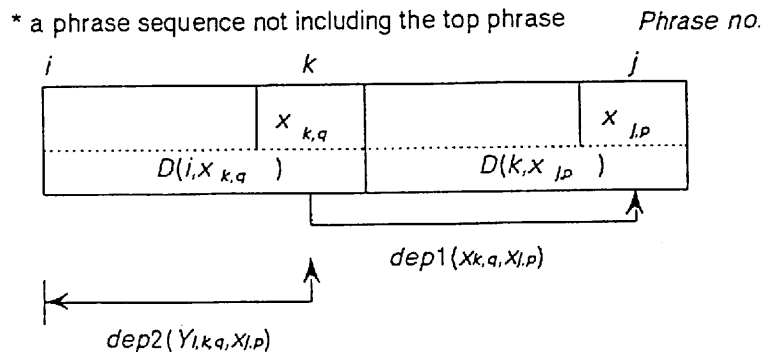
where $1 \leq j \leq N$, $1 \leq p \leq M$, N is the number of input phrases, M is the maximum number of recognition candidates for each uttered phrase, $x_{j,p}$ is a candidate of the j -th input phrase with the p -th best likelihood, and $c(x_{j,p})$ is its log-likelihood. A phrase sequence with one phrase candidate for each i -th to j -th input phrase and whose last phrase is $x_{j,p}$ is denoted by $X_{i,j,p}$. $Y_{i,j,p}$ is one of the dependency structures of $X_{i,j,p}$, and $w_{i,j-1}$ is the set of phrases that modify $x_{j,p}$ in

the sequence $X_{i,j,p}$. Here, $\text{dep}(w, x | Y)$ is the linguistic certainty factor of dependency relationships between w and x taking Y into account. The first item of the term on the right in Eq.(1) is the summation of phonetic likelihoods of the hypothesized sentence composed of its phrase sequence, and the second item is the summation of linguistic certainty factor. Maximizing Eq.(1) gives the sentence and its dependency structure as the speech recognition result.

To solve Eq.(1) effectively, a fast parsing algorithm using breadth-first search and beam search was used⁽³⁾. This algorithm is based on the fundamental algorithm⁽¹⁾. Although it offers sub-optimal solutions, it is practical because it requires much less processing



$$S(1, x_{j,p}, r) = r^{\text{th}} \max_{k,q,r1,r2} [S(1, x_{k,q}, r1) + D(k, x_{j,p}, r2) + \text{dep1}(x_{k,q}, x_{j,p}) + \text{dep2}(Y_{1,j,p}, x_{j,p})]$$



$$D(i, x_{j,p}, r) = r^{\text{th}} \max_{k,q,r1,r2} [D(i, x_{k,q}, r1) + D(k, x_{j,p}, r2) + \text{dep1}(x_{k,q}, x_{j,p}) + \text{dep2}(Y_{i,k,q}, x_{k,q})] \quad (i \neq 1)$$

Fig. 3 Illustration of deriving the recurrence relation among the objective function $i \leq k \leq j-1$, $1 \leq q \leq M$, and $1 \leq r, r1, r2 \leq L$. $r, r1$ and $r2$ indicate the beam ranks, L is the maximum number of beams, $S(1, x_{j,p}, r)$ and $D(i, x_{j,p}, r)$ are the r -th value of the element whose phrase sequence is $X_{i,j,p}$, and the dependency structure is $Y_{i,j,p}$. Here, $r^{\text{th}} \max[\cdot]$ is a function for deriving the r -th best value.

than the depth-first search⁽¹²⁾ that derives the optimal solutions.

The breadth-first algorithm is formulated as follows. First, $dep(w, x|Y)$ can be divided into two terms:

$$\begin{aligned} & dep(w_{1,j-1}, x_{j,p} | Y_{1,j,p}) \\ &= \sum_{x \in w_{1,j-1}} dep1(x, x_{j,p}) + dep2(Y_{1,j,p}, x_{j,p}) \quad (2) \end{aligned}$$

where $dep1$ is the certainty factor associated with dependency relationships of only the modifier and modificand phrases, and $dep2$ is the certainty factor associated with $Y_{1,j,p}$.

This algorithm uses beam-search to recursively derive $S(1, x_{j,p})$, the objective function's value of a phrase sequence from the top phrase to phrase $x_{j,p}$, as well as $D(i, x_{j,p})$, the value of a phrase sequence not including the top phrase ($i \neq 1$) as shown in Fig. 3. In these equations, each dependency structure for evaluation of $dep2$ is different. When the equations are calculated, $Y_{i,j,p}$ is stored for later use evaluating $dep2$. Initial values are given using phonetic likelihood based on HMMs.

After calculating the recurrence relation, the value

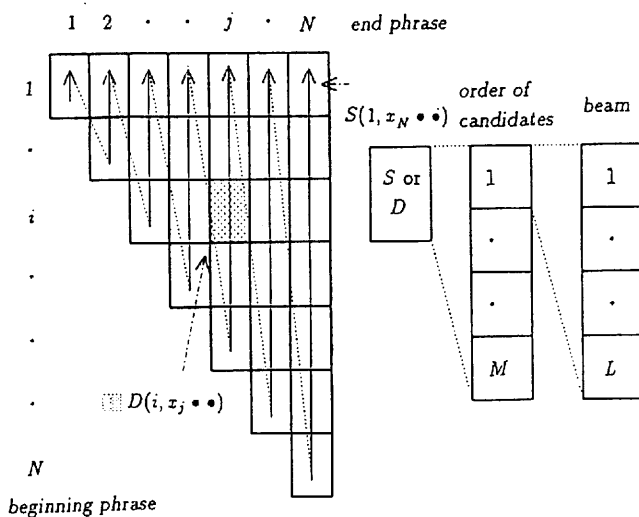


Fig. 4 Configuration of parsing table.

of the objective function is obtained:

$$T = \max_p [S(1, x_{N,p}, 1)] \quad (3)$$

where $1 \leq p \leq M$. The best sentence and its dependency structure are given through $Y_{1,N,p}$ where p maximizes Eq. (3). The parsing table is shown in Fig. 4. In Fig. 4, the first row corresponds to S , and others correspond to D . The phrase sequence for the first to N -th phrase corresponds to the right-most top cell. Each cell is composed of M sub-cells for the number of candidates, and each sub-cell is composed of L sub-cells for the beam width. The arrows show the sequence of calculating the recurrence relation. The processing amount order for this algorithm is $O(N^3 M^2 L^2)$.

4. Speech Recognition System-2

Figure 5 shows a block diagram of system-2. Inputs are sentence utterances. Phrase likelihood is calculated frame-synchronously using phoneme-based HMMs and transition networks. This calculation is based on the Viterbi algorithm, and yields several best phrase sequences for each frame whose last phrase is different from that of other sequences. The last phrase of each sequence for each end-frame is stored in lattice form. This lattice is reduced based on Viterbi phrase alignment. Then, the parser extracts the most likely sentence on the lattice.

4.1 Speech Recognition of Phrase Sequences

The syntactic constraints of sentences are governed

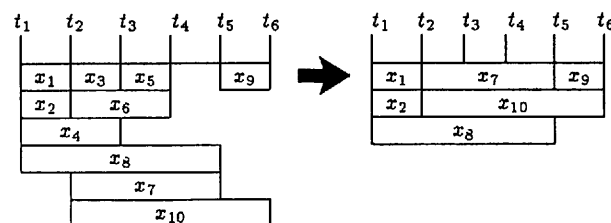


Fig. 6 Example of a reduced phrase lattice. x shows a phrase, and t shows a frame or time.

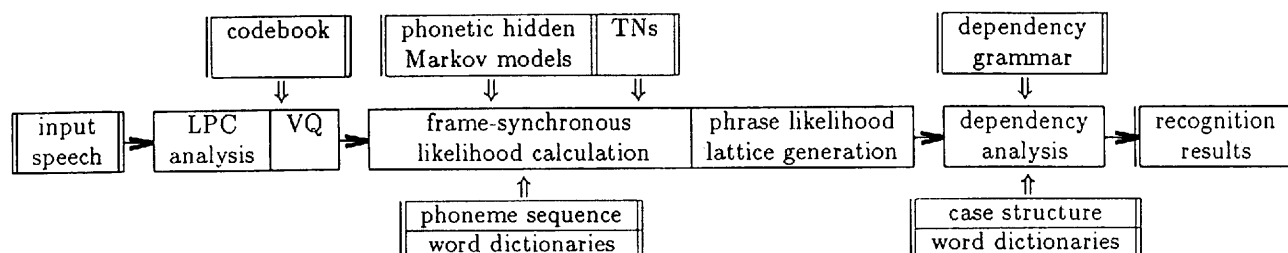


Fig. 5 Block diagram of the recognition system (system-2) for sentence speech.

by connection of phrase networks taking account of pauses between phrases. When the frame-synchronous parser parses the input from left to right using phoneme-based HMMs and networks based on the Viterbi algorithm, this calculation generates M -best phrase sequences whose last phrases are different from each other. The last phrases are stored in lattice form. In other words, there are M -phrases for each end-frame and the total number of phrase candidates in the lattice is approximately NM , if input is N -frame speech. After the whole lattice has been generated, significant phrase boundaries, which make consecutive phrase sequences are back-traced from the end of the speech to the beginning according to Viterbi phrase alignment. Phrase boundary candidates are then selected and a reduced lattice is generated. A simple example, where M is 2 and N is 6, is shown in Fig. 6. The left side of this figure is the original lattice. Three sets, that is (t_1, t_2, t_6) , (t_1, t_5, t_6) and (t_1, t_2, t_5, t_6) , can generate consecutive phrase sequences, and the new reduced lattice is on the right.

4.2 Dependency Parser for Sentence Speech

Consecutive phrases, whose number is H , of a sentence on the phrase lattice are given as follows.

$$x_{j_1, l_1, p_1} \cdots x_{j_H, l_H, p_H} \cdots x_{j_H, l_H, p_H}$$

where $1 \leq h \leq H$, $1 \leq j \leq N$, $1 \leq p \leq M$, N is the number of frames of the input sentence, and M is the number

of phrase candidates of each segment. Here, j_h, l_h, p_h is abbreviated to j, l, p . The term $x_{j, l, p}$ is a candidate of the $j-l$ -th to j -th frame with the p -th best likelihood. The other terms shown in Sect. 3.3 are also transformed by adding the subscript concerned with phrase length l .

Lattice parsing selects the best phrase sequence in the lattice taking account of HMM likelihood and linguistic certainty factor. It is equivalent to solving the following objective function:

$$T = \max_{H, (p)} \left[\sum_{h=1}^H c(x_{j, l, p}) + \max_Y \sum_{h=1}^H dep(w_{1, j}, x_{j, l, p} | Y_{1, j, l, p}) \right] \quad (4)$$

The breadth-first algorithm, which was described in Sect. 3.3, has been extended to sentence speech like other algorithms^{(13), (14)}. Our original algorithm shown in Table 2 uses a recurrence relation, and deals with the linguistic certainty factor associated with dependency structure. This is different from the other algorithms.

After calculating the recurrence relation, the value of the objective function is obtained:

$$T = \max_{l, p} [S(1, x_{N, l, p}, 1)] \quad (5)$$

where $1 \leq p \leq M$ and $1 \leq l \leq N$. The best sentence and its dependency structure are given through $Y_{1, N, l, p}$ where p and l maximize Eq. (5). Although the processing amount order for this algorithm is $O(N^5 M^2 L^2)$, it is not practical.

Table 2 Original parsing algorithm for sentence speech.

{1} : Loop for the end frame of the end phrase of the partial sequence
DO {2} to {6} for $j = 2, 3, \dots, N$
{2} : Loop for the beginning frame of beginning phrase
DO {3} to {6} for $i = j - 1, \dots, 1$
{3} : Loop for the length of end phrase
DO {4} to {6} for $l = 1, \dots, i$
{4} : if $j - i = l$, set the initial value
{5} : Loop for the end phrases
DO {6} for $p = 1, 2, \dots, M$
{6} : Computation of recurrence relation
{ Loop for the end phrase of the former sequence }
{6-1} : DO {6-2} to {6-5} for $k = i + 1, \dots, j - l$
{6-2} : DO {6-3} to {6-5} for $g = 1, \dots, k - i$
{6-3} : DO {6-4} to {6-5} for $q = 1, 2, \dots, M$
{ Loop for the beam width }
{6-4} : DO {6-5} for $r1 = 1, 2, \dots, L$
{6-5} : for $r2 = 1, 2, \dots, L$
* Evaluation of $S(1, x_{j, l, p}, r)$ or $D(i, x_{j, l, p}, r)$ taking account of $Y_{1, j, l, p}$ or $Y_{i, k, g, q}$
* Store of $Y_{i, j, l, p}$
{7} : Acquisition of the parsing results
* Detection of values, p and l , maximizing Eq.(5)
* Acquisition of the phrase sequence and its dependency structure using $Y_{1, N, l, p}$

1 or i	$k-g$	k	$j-l$	j
				frame
		$x_{k, g, q}$		$x_{j, l, p}$
$S(1, x_{k, g, q})$ or $D(i, x_{k, g, q})$			$D(k, x_{j, l, p})$	

4.3 Further Reduction of Processing Based on Viterbi Phrase Alignment

After selection of phrase boundaries, the reduced phrase lattice is generated. In this case, the parsing algorithm is extended as follows. Loops {3} for phrase length N and {5} for the number of fixed-length phrase candidates M in Table 2 are reduced to one loop for the number of free-length candidates $\tilde{M}$. In the same way, loops {6-2} and {6-3} are reduced to a loop. Then, the processing amount order for this extended algorithm is $O(\tilde{N}^3 \tilde{M}^2 L^2)$. If $\tilde{N} = N/10$, $\tilde{M} = M$, and the length of the input sentence is about 310 frames, then the parser reduces computation to approximately 10^{-8} of the amount for the original parser proposed in Sect. 4.2. The parsing table is similar to that of Fig. 4. Each cell in the parsing table is composed of $\tilde{M}$ sub-cells for the number of candidates, and each sub-cell is composed of L sub-cells for the beam-width.

5. Speech Recognition Experiments

Input utterances were sampled at a rate of 12 kHz. One frame was extracted every 10 ms with a 30 ms

Hamming window and converted into 34 acoustic feature parameters: power, 16 LPC cepstra, Δ power, and 16 Δ LPC cepstra.

In the training process, 216 phonetically balanced words were used. These utterances were manually labeled using 25 phoneme symbols including silence. Each phoneme was modeled by HMM and had 4 states and 7 transition paths. The parameter sequence was converted into a vector code sequence and a vector codebook composed of 256 prototype vectors was generated. Each HMM was trained using the forward-backward algorithm, and the code sequence for training was cut out based on the phonetic labels. Output probabilities were floored after training.

In the testing process, acoustic feature parameters of the input were generated in the same manner. These parameters were converted into a code sequence using the same speaker's codebook generated in the training process.

Talker-dependent preliminary word recognition tests were carried out with 216 words uttered by ten speakers. The system attained a word recognition rate of 98.4%.

Recognition tests for system-1 were then performed on 100 sentences (including 668 phrases in an essay) uttered phrase-by-phrase by the same speakers. The word dictionary had 360 entries and the perplexity of the phrase syntax was 40. Certainty factors of dependency relationships were empirically determined through the analysis of technical literature. If the training set is 216 words, the system attained a phrase recognition rate of 80.8% using only the intra-phrase syntactic parser. The dependency parser increased this rate to 86.8% as shown in Table 3.

Finally, to evaluate system-2, sentence speech recognition was tested for 71 sentences (including 418 phrases) uttered by two speakers, one male and one female. In this experiment, 2 training sets were used: 216 phonetically balanced words, and combination of 216

words and 29 sentences. The maximum number of each phrase segment $\tilde{M}$ was 5, and the beam-width L was 8. Word dictionary, word perplexity, and certainty factors were the same as in the preceding tests. Table 4 shows that the dependency parser increased the phrase recognition rate of 69.2% to 83.1% and increased the sentence recognition rate of 11% to 40%, if the training set was the combination of 216 words and 29 sentences. These results show the effectiveness of semantic dependency analysis.

6. Conclusion

This paper described two Japanese continuous speech recognition systems using an intra-phrase transition network grammar and an inter-phrase dependency grammar. System-1 was tuned for sentences uttered phrase-by-phrase and system-2 for sentence utterances, to make the amount of computation practical. In system-1, the best sentence was efficiently determined from input utterances using a bi-directional network parser and a breadth-first dependency parser. In system-2, it was determined using a frame-synchronous parser and an extended dependency parser combined with the Viterbi phrase alignment technique. Where the perplexity of the phrase syntax is 40, system-1 and system-2 attained phrase recognition performance in the sentence by approximately 86.8% and 83.1%, showing the effectiveness of the two systems, especially the inter-phrase dependency grammar.

Further development is needed to refine the linguistic certainty factor based on a statistical method and phoneme-based models based on continuous HMMs that take context dependency into account.

Acknowledgement

The authors wish to express their appreciation to Prof. Masaki Kohda of the University of Yamagata and to Kiyohiro Shikano of NTT Laboratories for their invaluable discussions.

References

- (1) Ozeki K.: "A multi-stage decision algorithm to select optimum bunsetsu sequences based on degree of kakariuke-dependency", Trans. IEICE Japan, **J70-D**, 3, pp. 601-609 (March 1987).
- (2) Nakagawa S. and Ito T.: "Recognition of spoken Japanese sentences using mono-syllable units and backward kakariuke parsing algorithm", Trans. IEICE Japan, **J70-D**, 12, pp. 2469-2478 (Dec. 1987).
- (3) Matsunaga S.: "A breadth-first technique of dependency relationships analysis for minimal phrase speech recognition", Trans. IEICE Japan, **J71-D**, 10, pp. 1966-1976 (Oct. 1988).
- (4) Bahl L. R., et al.: "Further results on the recognition of a continuously read natural corpus.", Proc. ICASSP (1980).

Table 3 Speech recognition results (system-1, phrase-by-phrase speech).

data	216 words		668 phrases	
	training		test	
dependency analysis	-		without	with
recognition rate (top 1 [%])	99.9	98.4	80.8	86.8
(top 5 [%])	100	100	96.8	-

Table 4 Speech recognition results (system-2, sentence speech).

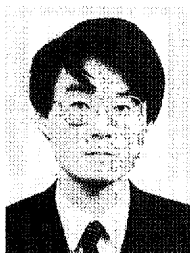
test data	phrase recognition rate [%]			
	71 sentences (418 phrases)			
HMM training data	216 words		216 words + 29 sent.	
dep. analysis	without	with	without	with
recog. rate(male)	45.5 (0)	55.3 (3)	63.6 (4)	78.9 (34)
recog. rate(female)	54.5 (11)	75.4 (32)	74.8 (18)	87.3 (46)
average	50.0 (6)	65.3 (18)	69.2 (11)	83.1 (40)

Each () indicates sentence recognition rate. [%]

- (5) Lee K-F. and Hon H-W.: "Large vocabulary speaker-independent continuous speech recognition using HMM", Proc. ICASSP, pp. 123-126, New York (1988).
- (6) Matsunaga S., et al.: "A continuous speech recognition system based on a two-level grammar approach", Proc. ICASSP, pp. 589-592, Albuquerque (1990).
- (7) Sagayama S. and Nakatsu R.: "Reversed DP matching applied to speech recognition of connected words", Proc. ASJ autumn meeting, 2-2-18, pp. 547-548 (Oct. 1979).
- (8) Sakoe H.: "Two-level DP-matching—A dynamic programming based pattern matching algorithm for connected word recognition", IEEE Trans. Acoust., Speech & Signal Process., ASSP-27, 6, pp. 588-595 (Dec. 1979).
- (9) Levinson S. E.: "Structural methods in automatic speech recognition.", Proc. IEEE, pp. 1625-1650 (Nov. 1985).
- (10) Bridle J., et al.: "An algorithm for connected word recognition.", Proc. ICASSP, pp. 899-902 (May 1982).
- (11) Sakoe H., et al.: "A high speed DP-matching algorithm based on frame synchronization, beam search and vector quantization.", Trans., IEICE Japan, J71-D, 9, pp. 1650-1659 (Sept. 1988).
- (12) Matsunaga S. and Kohda M.: "Speech recognition of minimal phrase sequence taking account of dependency relationships between minimal phrases", Trans. IEICE Japan, J70-D, 11, pp. 2102-2107 (Nov. 1987).
- (13) Ozeki K.: "A multi-stage decision algorithm to select optimum kakariuke structures from bunsetsu lattice", Trans. IEICE Japan, J70-D, 12, pp. 2621-2629 (Dec. 1987).
- (14) Kohda M.: "An algorithm for optimum selection of phrase sequence from phrase lattice" IECE Technical Report, NLC86-14 (Dec. 1986).



Sho-ichi Matsunaga received the B. S. and M. S. degrees in information science engineering from Kyushu University, Fukuoka, Japan in 1979 and 1981. Since 1981, he has been associated with Nippon Telegraph and Telephone Corporation, working on automatic speech recognition. Mr. Matsunaga is a member of the Acoustical Society of Japan.



Shigeru Homma was born in Tochigi Prefecture, Japan on November 18, 1958. He received the B. E. degree in electrical engineering, and the M. E. degree in information engineering from Tohoku University, Sendai, Japan in 1981 and 1983. From 1983 to 1987, he worked at the Yokosuka Electrical Communication Laboratories of Nippon Telegraph and Telephone Corporation in Kanagawa, Japan, where he did research on natural

language processing of Japanese. From 1987 to 1990, he worked at the NTT Human Interface Laboratories in Tokyo, Japan, where he was engaged in speech recognition research. Since 1990, he has been with the NTT Tohoku Software Plan and Produce Center in Sendai, Japan, where he is currently Head of the Development Section. Mr. Homma is a member of the Information Processing Society of Japan and the Acoustical Society of Japan.



Shigeki Sagayama was born on 12 May 1948 in Hyogo Prefecture, Japan. He received the B. S. and M. S. degrees in mathematical engineering and instrumentation physics from Tokyo University in Tokyo, Japan in 1972 and 1974. After joining the Nippon Telegraph and Telephone Public Corporation in 1974, he conducted research in speech analysis, speech recognition, and speech synthesis at the NTT Telecommunications Laboratories and, later, at NTT Human Interface Research Laboratories. Since 1990, he has been with the ATR Interpreting Telephony Research Laboratories in Seika-cho, Kyoto, doing speech recognition and speech synthesis research. Mr. Sagayama is a member of the Institute of Electrical and Electronics Engineers and the Acoustical Society of Japan.



Sadaoki Furui received the B. S., M. S., and Ph. D. degrees in mathematical engineering and instrumentation physics from Tokyo University, Tokyo, Japan in 1968, 1970, and 1978, respectively. After joining the Electrical Communications Laboratories, Nippon Telegraph and Telephone Corporation in 1970, he studied the analysis of speaker characterizing information in the speech wave, its application to speaker recognition as well as inter-speaker normalization and adaptation in speech recognition, the vector-quantization-based speech recognition algorithm, the spectral dynamic features for speech recognition, and the analysis of speech perception mechanism. He is currently the Executive Manager of Speech and Acoustics Laboratory at NTT Human Interface Laboratories. From December 1978 to December 1979 he was with the Staff of the Acoustics Research Department at Bell Laboratories, Murray Hill, New Jersey, as a visiting researcher working on speaker verification. Dr. Furui is a member of the Institute of Electrical and Electronics Engineers and the Acoustical Society of Japan. He received the Yonezawa Prize and the Paper Award from the IEICE in 1975 and 1988, respectively, and the Sato Paper Awards from the ASJ in 1985 and 1987, respectively. He received the Senior Award from the IEEE ASSP Society and the Achievement Award from the Minister of Science and Technology, both in 1989. He also received the Book Award from the IEICE in 1990. He is the author of "Digital Speech Processing, Synthesis, and Recognition" (Marcel Dekker, 1989) in English and "Digital Speech Processing" (Tokai University Press, 1985) in Japanese.