LINGUISTIC PROCESSING USING A DEPENDENCY STRUCTURE GRAMMAR FOR SPEECH RECOGNITION AND UNDERSTANDING

Sho-ichi MATSUNAGA
NTT Human Interface Laboratories
Musashino, Tokyo, 180, Japan

and

Masaki KOHDA
University of Yamagata
Yonezawa, Yamagata, 922, Japan

Abstract

This paper proposes an efficient linguistic processing strategy for speech recognition and understanding using a dependency structure grammar. The strategy includes parsing and phrase prediction algorithms. After speech processing and phrase recognition based on phoneme recognition, the parser extracts the sentence with the best likelihood taking account of the phonetic likelihood of phrase candidates and the linguistic likelihood of the semantic inter-phrase dependency relationships. A fast parsing algorithm using breadth-first search is also proposed. The predictor pre-selects the phrase candidates using transition rules combined with a dependency structure to reduce the amount of phonetic processing. The proposed linguistic processor has been tested through speech recognition experiments. The experimental results show that it greatly increases the accuracy of speech recognitions, and the breadth-first parsing algorithm and predictor increase processing speed.

1. Introduction

In conventional continuous speech recognition and understanding systems[1-4], linguistic rules for sentences composed of phrase sequences are usually expressed by a phrase structure grammar such as a transition network or context free grammar. In such methods, however, phoneme recognition errors and rejections result in incorrect transition states because of the strong syntactical constraints. These erroneous transitions cause the following candidates to be incorrectly chosen or the processing system to halt. Therefore, these errors and rejections can be fatal to speech understanding. Furthermore, a complete set of these grammatical rules for speech understanding is very difficult to provide.

To address these problems, this paper proposes a new linguistic processor based on a dependency structure grammar, which integrates a bottom-up sentence parser and a top-down phrase predictor. This grammar is more semantic and less syntactic than phrase structure grammar, and, therefore, syntactic positional constraint in a sentence rarely occurs with this parser. This effectively prevents extreme degradation in speech recognition from errors and rejections in phoneme recognition and greatly increases the accuracy of speech processing. This grammar only has two syntactic rules, so this parser is free of many cumbersome grammatical rules that are indispensable to other grammars. This grammar particularly suits phrase-order-free languages such as Japanese.

For the parser of this grammar, a depth-first parsing algorithm with backtracking which guarantees the optimal solution was devised[5]. However, parsing long sentences composed of many phrases with this algorithm can be time-consuming because of combinatorial explosion, since the amount of computation is exponential order with respect to the number of phrases. Therefore, a fast parsing algorithm using breadth-first search and beam search was developed. This algorithm is based on fundamental algorithms[6,7] which only take account of the dependency relationships of the modifier and modificant phrases, and it handles higher linguistic or semantic processing such as case structure. The processing ability of this breadth-first algorithm is equivalent to that of the depth-first algorithm.

To effectively recognize speech, the amount of phonetic processing must be reduced through top-down prediction of hypotheses. However, top-down control using the principal dependency structure is impossible. To solve this problem, this novel phrase predictor was devised. This predictor pre-selects hypotheses for phoneme recognition using prediction rules, and then it reduces the amount of phonetic processing. Prediction rules are created by integrating connection rules and phrase dependency structures.

The effectiveness of this linguistic processing was ascertained through speech recognition experiments.

2. Linguistic processor

2. 1 Dependency structure grammar

This grammar is based on semantic dependency relationships between phrases. The syntactic rules satisfy the following two constraints. First, any phrase, except the last phrase of a sentence, can modify only one later phrase. Each modification, called a dependency relationship or dependency structure, can be represented by one arc. Second, modification arcs between phrases must not cross. These rules are illustrated in Fig. 1. In two unacceptable sentences, one sentence is unacceptable because one phrase modifies the former phrase, and

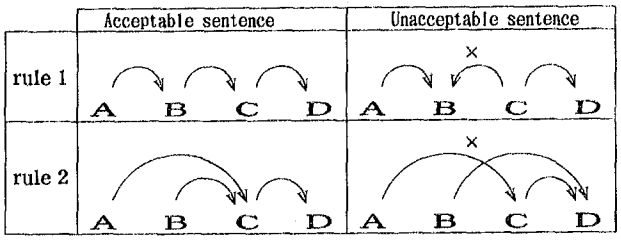


Fig. 1. Examples using a dependency structure grammar A,B,C and D are sentence phrases.

the other sentence is unacceptable because arcs cross in its dependency structures.

2. 2 Parser

After phonetic phrase recognition, recognition results are represented in a phonetic score matrix form as shown in Fig. 2. When analyzing dependency relationships, the parser extracts the most likely sentence in this matrix by taking into account the phonetic likelihood of phrase candidates and the linguistic likelihood of semantic inter-phrase dependency relationships. The parser also obtains the dependency structure that corresponds to the semantic structure of the extracted sentence.

2. 2. 1 Objective function

This parsing is equivalent to solving the following objective function using the constraints of dependency structure grammar. For simplicity, the following linguistic formulation is described for speech uttered phrase by phrase. The process for sentence speech is described in section 4.

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

where $1 \leq j \leq N$, $1 \leq p \leq M$, N is the number of input phrases, M is the maximum number of phonetic recognition candidates for each phrase, $x_{j,p}$ is a candidate of the j-th input phrase with the p-th best phonetic likelihood, and $c(x_{j,p})$ is its phonetic likelihood (positive value). Also, $X_{i,j,p}$ is a phrase sequence with one phrase candidate for each i-th to j-th input phrase and whose last phrase is $x_{j,p}$. $Y_{i,j,p}$ is one of the dependency structures of $X_{i,j,p}$, $w_{i,j-1}$ is the set of phrases that modify $x_{j,p}$ in the sequence $X_{i,j,p}$. Here, $dep(w,x|Y)$ is the linguistic likelihood (negative value) of dependency relationships between w and x taking Y into account. Namely, 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 likelihood. Maximizing Eq. (1) gives the sentence and the dependency structure of it as speech recognition and understanding results.

Because dependency structure grammar is compatible with case grammar[8], the linguistic semantic likelihood(dep) of the dependency structure is easily provided using case structure. The following are examples of items for evaluating dependency relationships: the disagreement between the semantic primitives of the modifier and that requested by the modificant, the lack of the obligatory case phrase requested by the modificant, and the existence of different phrases with the same case and modifying the same phrase. The likelihood values for these items are given heuristically.

To solve equation (1), a fast parsing algorithm using breadth-first search and beam search was developed. This algorithm deals with higher linguistic or semantic processing such as the case structure. Although this algorithm offers sub-optimal solutions, it is practical because it requires less processing than depth-first search.

2. 2. 2 Breadth-first parsing algorithm

The breadth-first algorithm is formulated as

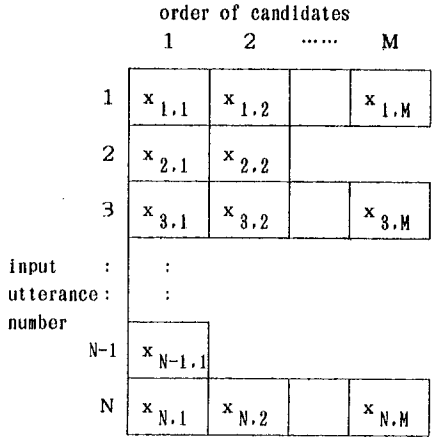


Fig. 2. A matrix of phrase candidates

follows.

First, $dep(w,x|Y)$ can obviously be divided into two terms.

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

where dep1 is the likelihood associated with dependency relationships of only the modifier and modificant phrases, and dep2 is the likelihood associated with $Y_{1,j,p}$. An example of dependency relationships is shown in Fig. 3.

Eqs. (1) and (2) give the objective function's value $S(1,x_{j,p})$ of a phrase sequence including the top phrase to $x_{j,p}$ in the sentence as:

$$S(1,x_{j,p}) = \sum_{h=1}^{j} c(x_{h,p}) + \sum_{h=1}^{j} \sum_{x \in w_{1,h-1}} dep1(x, x_{h,p}) + \sum_{h=1}^{j} dep2(Y_{1,h,p}, x_{h,p}) \quad (3)$$

On the other hand, the value of a phrase sequence not including the top phrase ($i \neq 1$) is defined as:

$$D(i,x_{j,p}) = \sum_{h=i}^{j} c(x_{h,p}) + \sum_{h=i}^{j} \sum_{x \in w_{i,h-1}} dep1(x, x_{h,p}) + \sum_{h=i}^{j-1} dep2(Y_{i,h,p}, x_{h,p}) \quad (4)$$

The main difference between Eqs. (3) and (4) is that

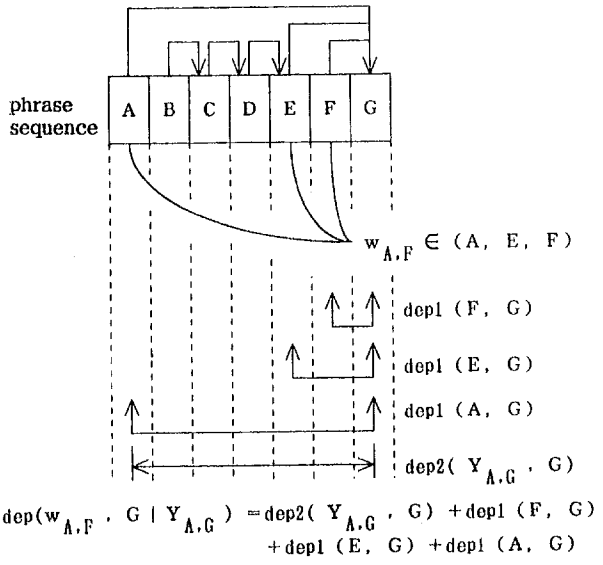


Fig. 3. Illustration of dependency relationships

$dep2(Y_{i,j,p},x_{j,p})$ is not evaluated in Eq. (4).

Using notation S and D, the recurrence relation among the objective functions are derived. This is shown in Fig. 4. The recurrence relation are transforms into the following equations using beam search.

$$S(1,x_{j,p},r) = \underset{k,q,r1,r2}{r^{th}\text{-}max}[S(1,x_{k,q},r1) + D(k+1,x_{j,p},r2) + dep1(x_{k,q},x_{j,p}) + dep2(Y_{1,j,p},x_{j,p}), \quad \text{if } i=1 \quad (5')$$

$$D(i,x_{j,p},r) = \underset{k,q,r1,r2}{r^{th}\text{-}max}[S(i,x_{k,q},r1) + D(k+1,x_{j,p},r2) + dep1(x_{k,q},x_{j,p}) + dep2(Y_{i,k,q},x_{k,q}), \quad \text{if } i\neq 1 \quad (6')$$

where $i\leq k\leq j-1$, $1\leq q\leq M$, and $1\leq r, r1, r2\leq L$. Here, r, r1 and r2 indicate the rank of beam, 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, $r^{th}\text{-}max[\ ]$ is a function for deriving the r-th best value. When Eq. (5') or (6') is calculated, $Y_{i,j,p}$ is stored for use in the later stage of evaluating dep2.

Initial values are given as follows.

$$S(1,x_{1,p},1) = c(x_{1,p}) + dep2(Y_{1,1,p},x_{1,p}), \quad \text{if } i=1 \text{ (top phrase)} \quad (7)$$

$$D(i,x_{i,p},1) = c(x_{i,p}), \quad \text{if } i\neq 1 \text{ (not top phrase)} \quad (8)$$

After calculating the recurrence relation, the value of the objective functions is obtained,

$$T = \max_{p}[S(1,x_{N,p},1)], \quad (9)$$

where $1\leq p\leq M$. The best sentence and its dependency structure are given through $Y_{1,N,p}$ where p maximizes Eq. (9). The parsing table is shown in Fig. 5 and the parsing algorithm is shown in Table 1. In Fig. 5, the first row corresponds to S, and others correspond to D. The phrase sequence for first to N-th phrase corresponds to the right-most top cell. Each cell is composed of ML sub-cells. Arrows show the sequence of calculating the recurrence relation. The processing amount order for this algorithm is $O(N^3M^2L^2)$.

Comparing the theoretical amount of processing for these two parsing algorithms, the breadth-first parsing algorithm clearly requires much less processing than the depth-first parsing algorithm. The amount of processing for each parsing algorithm is shown in Fig. 6.

2. 3 Predictor

To pre-select the phrase hypotheses for the speech recognition, the predictor is devised[9], using prediction rules created by integrating connection rules and dependency structures of phrases. These rules are described with rewriting rules:

$(X_{i,j})\text{->}(X_{i,k})(X_{k+1,j})$

where $X_{i,j}$ is the phrase sequence for the i-th to j-th phrase. $(X_{i,j})$ is the sequence with a closed-dependency-structure where the tail phrase x_j has the dependency relationships with phrases out of $X_{i,j}$, and other phrases in $X_{i,j}$ have dependency relationships with phrases within $X_{i,j}$. $(X_{i,j})$ is divided into two phrase sequences with the closed-dependency-structure by modifying x_k by x_j, and following $X_{i,k}$ by $X_{k+1,j}$. A single phrase x_i is also

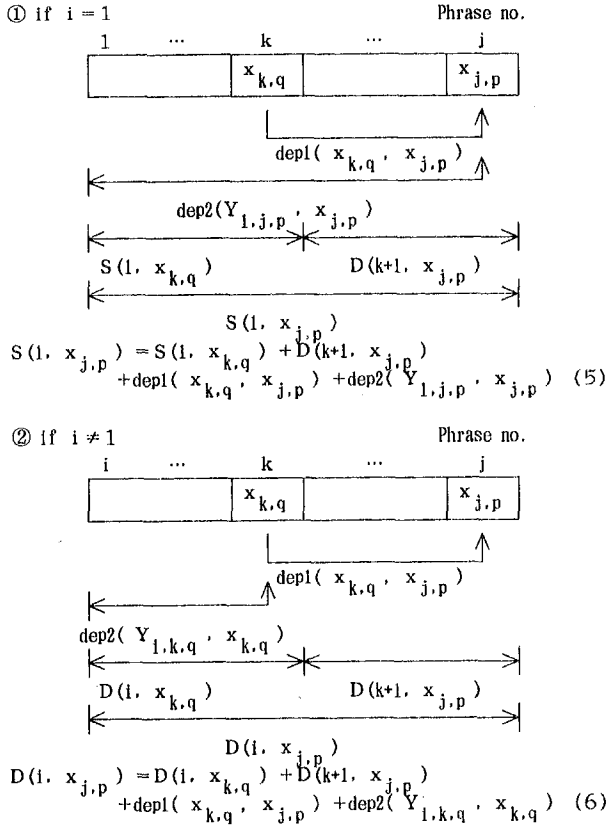


Fig. 4. Illustration of deriving recurrence relation among objective functions

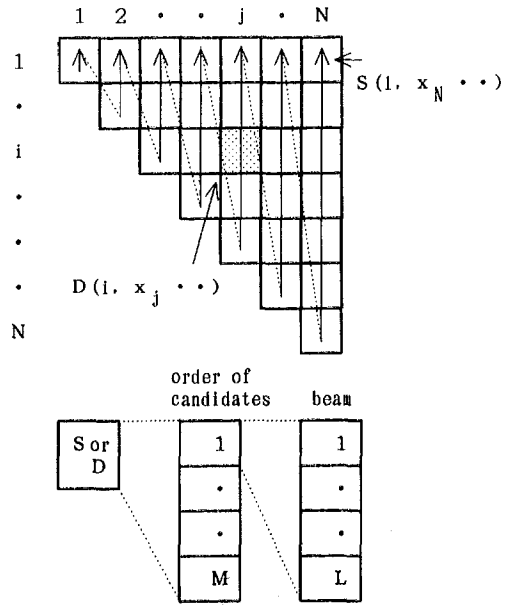


Fig. 5. Configuration of a parsing table

regarded as a phrase sequence with a closed-dependency-structure. These rules are described for the sequence, the i-th phrase to j-th phrase modified by the i-th phrase, as

$(X_{i,j})\text{->}(x_i)(X_{i+1,j})$

The hypotheses are predicted as follows.

<1> x_i is detected as a reliable phrase recognition result. If there are no reliable phrase candidates in

the i-th phrase recognition results, the following procedure is not carried out.

<2> The rules whose left term is scanned are such as

$(X_{i+1,j})\text{->}(X_{i+1,k})(X_{k+1,j})$

After the left-most derivation is repeated to detect hypotheses for i+1-th phrase speech recognition, x_{i+1} is detected in the following form.

$(X_{i+1,j})\text{->}(x_{i+1})(X_{i+2,h})\text{---}(X_{k+1,j})$

Generally, there is more than one $(X_{i+1,j})$, so $\mathbf{x}_{i+1}$ is a set of phrases.

<3> The phrase recognition is carried out for the i+1-th phrase utterance whose hypotheses are elements of the set $\mathbf{x}_{i+1}$.

<4> If the reliable phrase recognition result was detected in operation <3>, the rules which derived elements of $\mathbf{x}_{i+1}$ are scanned again and hypotheses for the next utterance are derived using same procedure as <2>.

<5> These operations, namely hypotheses derivation and its phonetic verification, are carried out until x_j is detected.

<6> The detected phrase sequence $X_{i,j}$ and its dependency structure $Y_{i,j}$ is passed to the parser.

During these operations, if the phrase recognition results are unreliable in operation <3>, the detection process of $X_{i,j}$ is halted and phrase recognition for all hypotheses is carried out.

Although Japanese is a phrase-order-free language, there are some relatively fixed phrase-order parts in a sentence. These rules are applied to these parts. The number of hypotheses and the amount of acoustic processing can thus be reduced, maintaining the above characteristics of the dependency structure grammar. By linking the predictor to the parser, parsing can be accomplished using the dependency structures detected in operation <6> of the prediction procedure. This linkage method greatly increases parsing speed.

3. Speech recognition experiments

3. 1 Speech recognition system

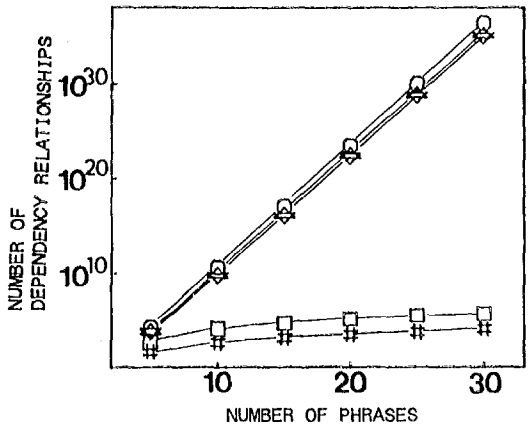
The speech recognition and understanding system is shown in Fig. 7. The system is composed of acoustic and linguistic processing stages. The acoustic processing stage consists of a feature extraction part and a phoneme recognition part[10,11]. The linguistic processing stage consists of a phrase recognition part[11], a parsing part (a dependency relationship analysis part), and a phrase prediction part. The linguistic processing stage uses a word dictionary, word connection rules for intra-phrase syntax, dependency relationships rules and phrase prediction rules. The word dictionary is composed of pronunciation expressions, parts of speech and case structures. Dependency relationship rules produce negative evaluation values that are set to the dependency relationships contrary to case structure discipline.

3. 2 Speech recognition process

For separately uttered phrases, acoustic feature parameters are extracted and bottom-up phoneme recognition is carried out. The phrase hypotheses for top-down phoneme recognition are pre-selected by the predictor. The pre-selection is also carried out using bottom-up phoneme recognition results[12]. Next, top-down phoneme verification is carried out and phrase recognition results are generated. Phrase recognition results are represented in the form of score matrix with phonetic recognition scores averaged for each hypothesized phrase. When the end of a sentence is detected, the parser extracts the phrases with the best sentence likelihood by scanning this matrix, and determines the dependency structure of the extracted phrases.

Table 1. Parsing algorithm

{1} Loop for the end phrase of the partial sequence
DO {2} to {5} for $j = 1,2,\text{---},N$
{2} Loop for the candidate
DO {3} to {5} for $p = 1,2,\text{---},M_j$
{3} Setting the initial value
SET $S(1,x_{1,p},1)$ or $D(j,x_{j,p},1)$ (Eqs. (7),(8))
If $j = 1$, go back to {2}.
{4} Loop for the beginning phrase of the partial sequence
DO {5} for $i = j\text{-}1,j\text{-}2,\text{---},1$
{5} Calculation of reccurence relation
< *Loop for the end phrase of the former sequence* >
{5-1} DO {5-2} to {5-4} for $k = i,i+1,\text{---},j\text{-}1$
{5-2} DO {5-3} to {5-4} for $q = 1,2,\text{---},M_k$
< *Loop for the beam width* >
{5-3} DO {5-4} for $r1 = 1,2,\text{---},L$
{5-4} DO for $r2 = 1,2,\text{---},L$
* Evaluation of $S(1,x_{1,p},r)$ or $D(j,x_{j,p},r)$ taking account of $Y_{i,j,p}$ or $Y_{i,k,q}$ (Eqs. (5'),(6'))
* Store of $Y_{i,j,p}$
{6} Acquisition of the parsing results
* Detection of value p maximizing Eq. (9)
* Acquisition of the phrase sequence and its dependency structure using $Y_{1,N,p}$



○ enumerative method
△ depth-first method
▽ breadth-first method(L=∞)
□ breadth-first method(L=8)
breadth-first method(L=1)

Fig. 6. Comparison of processing amount

3. 3 Performance

The effectiveness of the proposed linguistic processor was tested in speech recognition experiments. The experiments were carried out using 20 sentences containing 320 phrases uttered by 2 male speakers. These results are shown in Table 2.

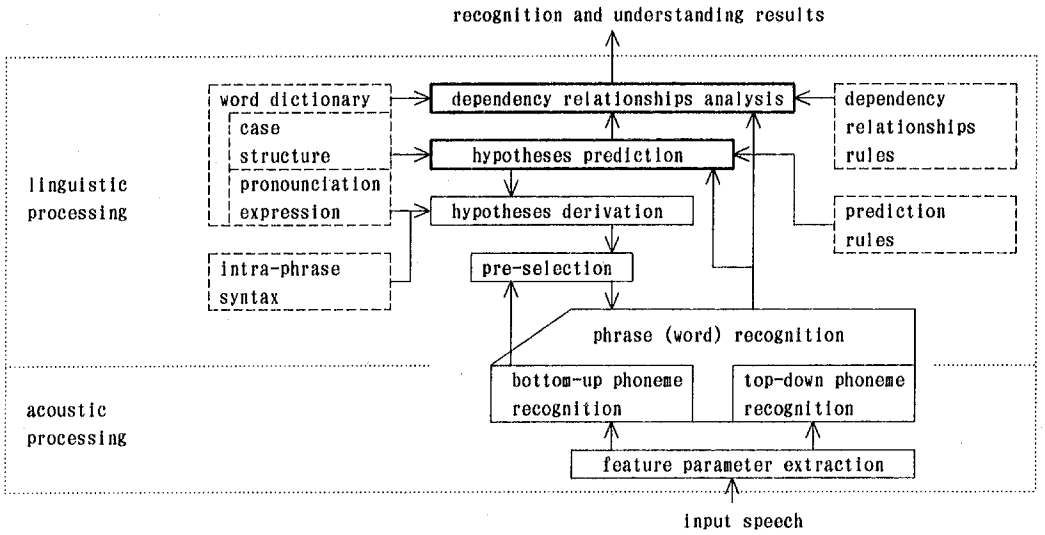
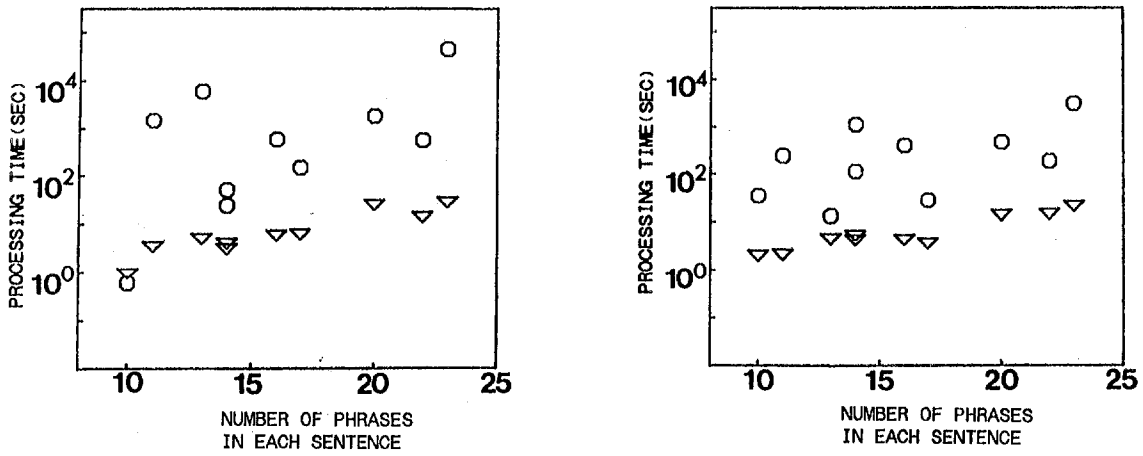


Fig. 7. Speech recognition and understanding system



(a) speaker KI
○ depth-first method
▽ breadth-first method(L=8)

(b) speaker KS
○ depth-first method
▽ breadth-first method(L=8)

Fig. 8. Comparison of processing time for dependency relationships analysis

Table 2. Speech recognition results

	phonetic recognition*		without parser	with parser			
			phrase recognition rate [%]	depth-first parsing		breadth-first parsing	
	number of hypotheses	phonetic processing time	(): within top 8 candidates	phrase recognition rate [%]	parsing time	phrase recognition rate [%]	parsing time
without predictor	542	1.0	57 (87)	77	1.	77	1.
with predictor	169	0.48	60 (89)	78	0.07	78	0.89

* *Recognition of predicted phrases (33% to the total input phrases)*

The proposed parser using the depth-first parsing algorithm increased phrase recognition rate by approximately 20% (from 57% without the parser to 77% with the parser). This result shows the effectiveness of a parser using a dependency structure grammar.

The processing time with the breadth-first algorithm was reduced to approximately 1% of that with the depth-first algorithm for sentence parsing, while keeping the same level of speech recognition rate as with the depth-first algorithm. This result shows the great effectiveness of the breadth-first parsing algorithm. This result is shown in Fig. 8 for each speaker when M is 3 and L is 8.

Next, using 26 rules, the prediction was carried out for 33% of the total input phrases. It reduced acoustic processing time to 60% at these parts in a sentence, and it increased speech recognition speed. Finally, linking the predictor to the parser reduced

parsing time to less than 10% of the time for the depth-first parser, and to approximately 90% of the time for the breadth-first parser. This shows the usefulness of the linkage.

4. Breadth-first parsing algorithm for sentence speech recognition

The breadth-first parsing algorithm for the sentence speech or connected phrase speech is devised[13] by the same procedure as in section 2. 2. Based on basic expansion algorithms[14,15] from phrase-wise to sentence speech, the speech recognition and understanding accuracy using the proposed algorithm is greatly increased compared to the accuracies using the basic algorithms. In the sentence speech, phrase recognition results after phonetic processing are represented in a score lattice form with phonetic recognition scores averaged. The parser extracts the best sentence composed of a phrase sequence by scanning this lattice. The processing order is $O(N^5M^2L^2)$, which is practical amount of computation, where N is the number of detected phrase boundaries in the uttered sentence, M is the maximum number of phonetic recognition candidates for each phrase segment from one boundary to the next boundary, and L is the maximum number of beams.

The effectiveness of this parser was tested through sentence speech recognition with one speaker uttering 10 sentences containing a total of 67 phrases. This parser increased phrase recognition performance in the sentences by approximately 49% (from 27% without the parser to 76% with the parser).

5. Conclusion

This paper proposed an efficient linguistic processing strategy for speech recognition and understanding using a dependency structure grammar. This grammar suits processing of phrase-order-free languages such as Japanese and processing the result of front-end speech recognition, which is usually erroneous. This linguistic processing strategy includes bottom-up parsing and a top-down phrase hypotheses predictor. In particular, the bottom-up parser, taking account of the phonetic and linguistic likelihood, greatly increases the accuracy and speed of speech recognition. The predictor reduces the amount of phonetic processing by pre-selecting the phrase hypotheses. The effectiveness of this parser and predictor was shown in speech recognition experiments.

Future development is include the statistical likelihood of dependency relationships, integration with the statistical phonetic method like Hidden Markov Models, and higher linguistic processing using the semantics and context knowledge.

Acknowledgment

The authors would like to express their appreciation to Dr. Kiyoshi Sugiyama and Dr. Sadaoki Furui, for their invaluable guidance. The authors would also like to thank to Dr. Kiyohiro Shikano and Shigeki Sagayama for their useful suggestions.

References

[1] Leser V.R., et al(1975)'Organization of the Hearsay II speech understanding system.',IEEE Trans. ASSP.,23,1,pp.11-24.

[2] Woods W.A.(1976)'Speech understanding system - Final Report',Tech. Rep.,3438.

[3] Levinson S.E.(1985)'Structural Methods in automatic speech recognition',Proceeding of the IEEE,11, pp.1625-1650.

[4] Ney H.(1987),'Dynamic programming speech recognition using a context-free grammar.'Proc. 1987 ICASSP, pp.69-72.

[5] Matsunaga S. & Kohda M.(1986)'Post-processing using dependency structure of inter-phrases for speech recognition.'Proc. Acoust. Soc. Jpn. Spring Meeting,pp.45-46.

[6] Hidaka T. & Yoshida S.(1983)'Syntax analysis for Japanese using case grammar.'Natural Language Processing Technique Symposium,pp41-46.

[7] Ozeki K.(1986)'A multi stage decision algorithm for optimum bunsetsu sequence selection.' Paper Tec. Group, IECE Japan,SP86-32,pp.41-48.

[8] Filmore C.(1968)'The case for case.' in Bach and Harms(eds.),1-88

[9] Matsunaga S. & Sagayama S.(1987)'Candidates prediction using dependency relationships for minimal phrase speech recognition.'Paper Tec. Group, IEICE Japan,SP87-29,pp.59-62.

[10] Aikawa K., Sugiyama M. & Shikano K.(1985)'Spoken word recognition based on top-down phoneme segmentation.'Proc.1985 ICASSP,pp33-36.

[11] Matsunaga S. & Shikano K.(1985)'Speech recognition based on top-down and bottom-up phoneme recognition.' Trans. IECE Japan,J68-D,9,1641-1648.

[12] Matsunaga S. & Kohda M.(1987)'Reduction of word and minimal phrase candidates for speech recognition based on phoneme recognition.'Trans. IEICE Japan,J70-D,3,pp.592-600.

[13] Matsunaga S.(1988)'Dependency relationships analysis for connected phrase recognition.' Nat. Conv. IEICE Japan,SA-1-3,pp.345-346.

[14] Ozeki K.(1986)'A multi-stage decision algorithm for optimum bunsetsu sequence selection from bunsetsu lattice.'Paper Tec. Group, IECE Japan, COMP86-47, pp.47-57.

[15] Kohda M.(1986)'An algorithm for optimum selection of phrase sequence from phrase lattice.' Paper Tech. Group, IECE Japan,SP86-72,pp.9-16.