STUDIA PHONOLOGICA IX (1975)

Continuous Speech Understanding System LITHAN

Toshiyuki SAKAI and Seiichi NAKAGAWA

Summary

We have developed LITHAN (LIsten-THink-ANswer) speech understanding system which automatically recognizes continuously uttered speech utilizing higher linguistic information such as syntactic, semantic, pragmatic information.

This system predicts possible words utilizing linguistic information at the unrecognized portion of the input utterances, and identifies the predicted word by using the optimum matching algorithm between a recognized phoneme string and the phoneme string of the word dictionary.

The system could parse sentences by tree searching, but the results of phoneme recognition and word identification are not always correct, therefore, we propose a new tree search method.

LITHAN uses many types of a priori information; the statistic of each phoneme; the similarity matrix between phonemes; the word dictionary; the spoken grammar with the additional information as regards the spoken grammar; the semantic and pragmatic information.

We have applied this efficient, flexible system to restricted utterances which include about 100 words used to perform operational command and query the status of a computer network. When tested on a sample of 200 sentences spoken by 10 male speakers at a normal speed, 64% of the sentences and 93% of the output words were recognized correctly.

1. Introduction

Man's primary natural method of communication is speech. Man-machine communication by speech would be very efficient and convenient. Therefore, many researchers have studied automatic speech recognition by machine.[1] As the results of their works, we find that automatic speech recognition on word-by-word basis except the case of very limited vocabulary is very difficult.

Recently, speech understanding systems (SUS) which understand and answer imput speech of a natural language, came to be studied particularly in U.S.A.[2][3] In general, a SUS is composed of various levels, each of which has the knowledge of its own. These levels are acoustic, parametric, lexical, sentence and semantic ones[2]. The levels have the statistic of each phoneme, the phoneme similarity

Toshiyuki SAKAI (坂井利之): Professor, Department of Information Science, Kyoto University.
Seiichi NAKAGAWA (中川聖一): Graduate course, Department of Electrical Engineering, Kyoto University, Kyoto, 606, Japan.

matrix, the word dictionary and the spoken grammar respectively as the given knowledge. A SUS uses synthetically as various information as possible, especially the information of higher levels, that is, the semantic and pragmatic information. For the purpose of efficient utilization of linguistic information, the input should be sentences which describe only a restricted world. This world is called a task.

We have developed LITHAN (LIsten-THink-ANswer) speech understanding system and applied this universal system to continuous speech of a natural language (Japanese). The task selected for this system is the operational command and query of the status of a computer network.

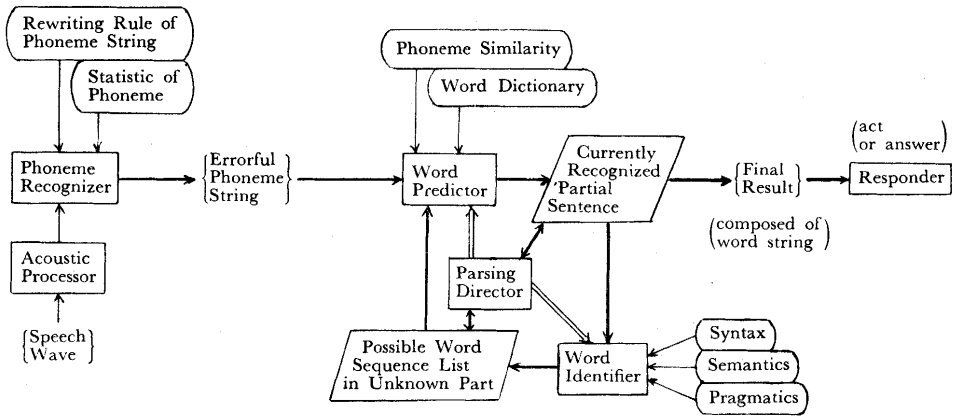

Fig. 1. Configulation of speech understanding system -LITHAN-.

Figure 1 indicates the block diagram of LITHAN, LITHAN is composed of Acoustic Processor, Phoneme Recognizer, Word Identifier, Word Predictor, Responder and Parsing Director.

Acoustic Processor has a 20-channel 1/4-octave filter-bank spectrum analyzer. Speech signal is passed into a pre-emphasis circuit with a slop of 6-dB per octave below 1600 Hz and fed into the 20-channel filter-bank. Its output is rectified, smoothed and sampled at every 10ms intervals, thus yielding a short time spectrum of 20 dimensions. The filters cover the frequency spectrum from 200 Hz to 6,400 Hz.

Phoneme Recognizer converts the time-series of the spectrum into a phoneme string, utilizing the statistic of phonemes and the rewriting rules of a phoneme string consulting to given knowledge. Word Identifier identifies a predicted word in the above mentioned recognized phoneme string and calculates the likelihood of the word. Word Identifier has also the phoneme similarity matrix and the word dictionary as the given knowledge.

Word Predictor predicts plausible words in the unknown portion of the input sentences. And these proposed words are identified by Word Identifier, utilizing

the syntactic, the semantic and the pragmatic information. Parsing Director directs Word Identifier passing it the words to be identified and the portion of the phoneme string to be matched against them, and builds up word strings based on the results of the identification. Parsing Director also directs Word Predictor passing it some word strings to be predicted. Responder (under development) tries to understand the meaning of the recognized sentence and answers a query or acts according to the input command.

At present, the vocabulary size of this task is 101: 21 predicates (mainly verb), 60 nouns, and 10 prepositions and others. But the word classification differs slightly from that of Japanese grammar. Some examples of input sentences are shown below. The equivalent English sentence is given in the parenthesis.

1. Keisanki cyuono zikidisuku sochi sanban kara keisanki gazoe deta yono rodoseyo. (Load the 4th datum from the 3rd magnetic disk device of central computer to imageprocessing computer.)

2. Keisanki hanyode zyobuwa ikutsu hashitteiruka? (How many jobs are being executed in general computer?)

3. Shiyosya Koga gozen zyunizi kara gogo ichizihan made keisanki gazono nizigenhyozi sochino yoyakyo suru. (User A reserves the X-Y plotter of image-processing computer from twelve o'clock in the morning to half past one in the afternoon.)

4. Keisanki kokanno zikitepu sochi kyubanni aoi zikitepu rokubano kakeyo. (Set the 6th blue magnetictape on the 9th magnetictape device of IMP computer.)

2. Phoneme Recognizer

Phoneme Recognizer segments input speech into a unit of phoneme (We call this unit a segment, that is, by a segment we mean a portion of the utterance which is hypothesized to be a single phoneme.), and assigns one of the phoneme categories to the unit. We classify Japanese phonemes into following categories.

1. vowel /a/, /i/, /u/, /e/, /o/
2. semi-vowel /y/, /w/
3. nasal /m/, /n/, /g̃/
4. voiced plosive /b/, /d/, /g/
5. liquid-like /r/
6. voiced fricative /z/, /dz/
7. voiceless plosive /p/, /t/, /k/
8. voiceless fricative /s/, /ʃ/
9. affricate /c/
10. aspirated /h/
11. syllabic nasal /N/
12. silence or pause /./, /-/, ///*

* in the order of increasing duration, where /./, /-/, /// may correspond to a buzz sound, a chocked sound and a pause of phrase boundary, respectively.

LITHAN treats /z/ and /dz/ or /s/ and /ʃ/ as the same phoneme. And LITHAN does not classify voiceless plosive group, therefore, we simply denote this group as /p/.

Figure 2 shows the block diagram of Phoneme Recognizer. Phoneme Re-

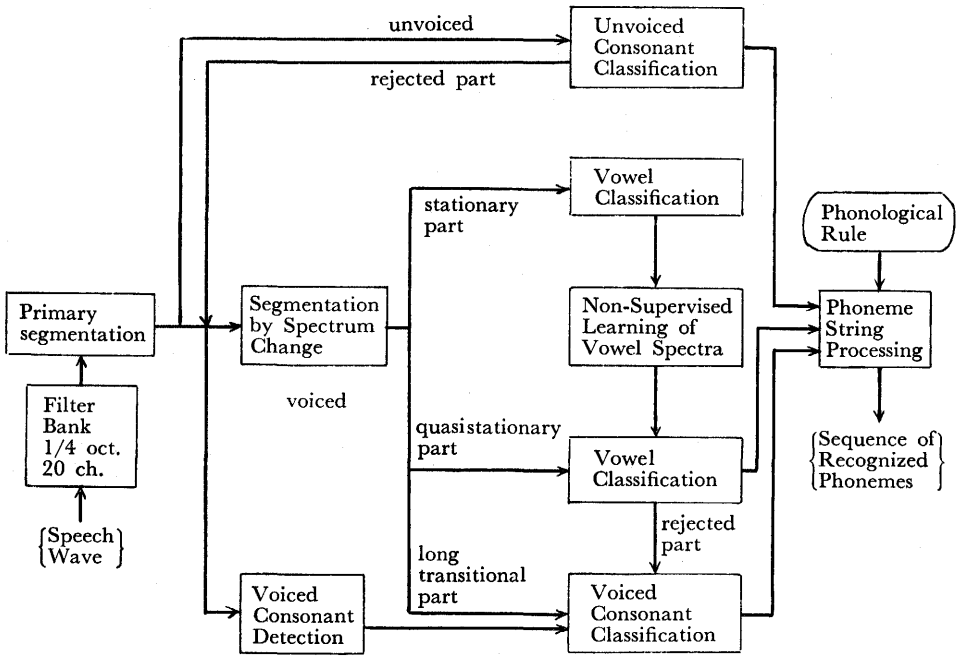

Fig. 2. Acoustic and phonemic processing.

cognizer performs primary segmentation, that is, classifies input speech into silence, voiceless-nonfricative (/p/, /h/), voiceless-nonplosive (/s/, /c/, /h/) or voiced group based on energy and deviation around the low or high frequency of (20-dimensional) spectrum every 10ms (We call this spectrum one frame hereafter.), etc. If a sequence of phonemes is composed of the same phonemes, they will be combined. On the other hand, if it is irregular, it will be smoothed. The output of this algorithm is a sequence of continuous and non-overlapping segments. The segment classified as the voiceless phoneme group is further classified into one of the detailed group corresponding to each phoneme on the basis of the segment duration, the presence of silence in preceding segment and spectra change, etc.

The segment classified as the voiced group is determined whether each frame is stationary or transient on the basis of the degree of spectra change between that frame and the preceding frame or the following frame.

The most stationary frames are used for non-supervised learning of vowels' spectrum patterns. A stationary part and a quasi-stationary part are regarded as a portion of vowels. The portion of voiced consonants is one of the followings: 1) a rejected portion by vowel recognition process; 2) a long transient portion; 3) a portion of weak energy with concave form. The spectral patterns of voiced consonants are gradually trained by using the learned ones of vowels.

The recognition of vowels and voiced consonants is based on Bayes' descri-

minant functions. Semi-vowels are recognized by using the rewriting rules for the errorful phoneme string. The result of each segment consists of the first candidate of phonemes, the second candidate, the degree of confidence (reliability) of the first candidate and the segment duration. Then, this recognized phoneme string is corrected by the phoneme rewriting rules or the phonological rules.

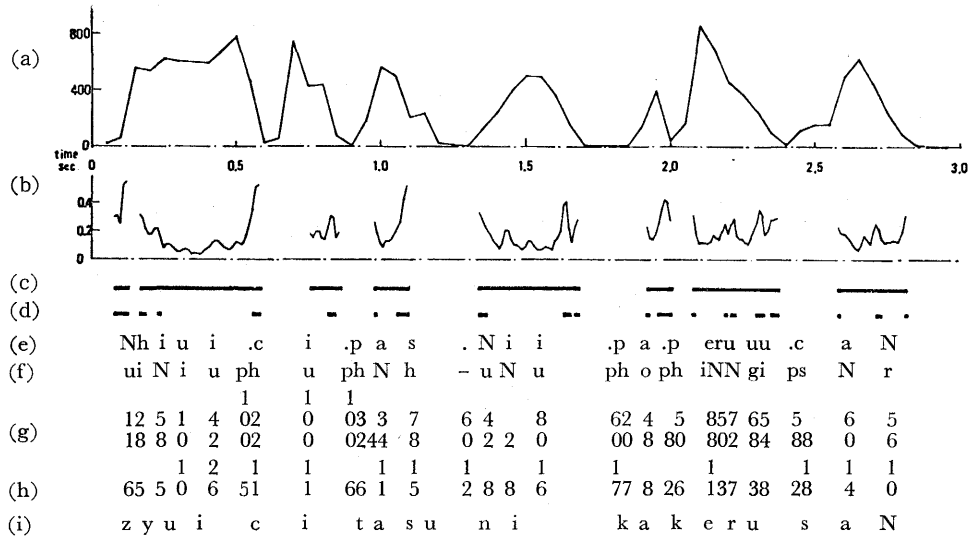

Fig. 3. An example of phoneme recognition result.
(a): power of input speech, (b): degree of spectra change,
(c): voiced part, (d): transitional part, (e): first candidate,
(f): second candidate, (g): confident degree of first condidate ($\times 100$),
(h): segment duration ($\times 10$ ms), (i): input speech (arithmetic expression: $11+2\times 3$)

Figure 3 indicates an example of each process mentioned above. The energy (power) is defined as the root mean square of the (20-dimensional) spectrum in given the frame. Acoustic Processor produces 10-bits samples (thus given 10×20 bits/10ms$=$20,000 bits/sec). When the value of confident degree (reliability) of the first candidate is equivalent to 0, it means that the reliability of first candidate is much as same as that of second one. Because the word "tasu" (plus) was devocalized, it was recognized as "pas".

3. Word Identifier

Word matching is defined as each phoneme of a recognized phoneme string having one to one corresponding to each phoneme of a word in the dictionary. To evaluate this matching, we introduce a concept of the similarity between two phonemes. Since the phoneme recognition is performed by using the statistic (means and covariance) of the spectrum (20 dimensions) for each phoneme, if Phoneme Recognizer made mistakes concering the phoneme recognition, we can consider that their errors were caused by the spectrum distribution of a correct

and a misrecognized phoneme being very similar. These errors are generally divided into three kinds, that is, a) substitution error; b) insertion error; c) omission error. Figure 4 shows examples of three type errors.

We obtained the phoneme similarity S(i, j) for all pairs of phonemes (/i/, /j/) by the distance between the spectrum distributions of the two phonemes (/i/, /j/). Therefore we can evaluate the dégree of the matching between two pho-

Table 1. Phoneme similarity matrix.

In \ Out	a	i	u	e	o	N	y	w	m	n	ɡ̃	b	d	g	r	z	s	c	h	p	t	k	.	–	/
a	100	36	51	69	75	65	70	82	51	55	58	50	51	40	63	36	5	5	70	5	5	5	5	5	5
i	36	100	83	73	56	85	82	40	72	74	85	79	76	77	75	69	5	30	70	50	50	50	5	5	5
u	51	83	100	74	80	89	78	70	81	83	89	88	81	79	82	72	5	5	5	5	5	5	5	5	5
e	69	73	74	100	69	73	92	61	59	69	74	67	72	63	78	62	5	5	5	5	5	5	5	5	5
o	75	56	80	69	100	80	70	91	64	65	75	75	66	60	74	50	5	5	5	5	5	5	5	5	5
N	65	85	89	73	80	100	75	72	84	85	84	81	74	69	78	69	5	5	5	5	5	5	5	5	5
y	75	87	83	95	75	80	100	66	68	73	82	76	77	66	87	67	5	30	50	50	50	50	5	5	5
w	87	45	75	66	96	77	66	100	62	61	71	70	61	50	75	38	5	5	5	50	50	50	5	5	5
m	61	82	91	69	74	94	68	62	100	92	86	85	75	68	83	55	5	5	70	5	5	5	5	5	5
n	65	84	93	79	74	95	73	61	92	100	88	86	84	71	87	69	5	5	70	5	5	5	5	5	5
ɡ̃	68	95	99	84	85	94	82	71	86	88	100	93	89	87	89	79	5	30	70	85	85	85	30	5	5
b	60	89	98	77	85	91	76	70	85	86	93	100	90	84	89	72	5	50	50	85	85	85	40	5	5
d	61	86	91	82	76	84	77	61	75	84	89	90	100	85	88	87	5	50	50	85	85	85	40	5	5
g	50	87	89	73	70	79	66	50	68	71	87	84	85	100	75	78	5	50	70	85	85	85	40	5	5
r	73	87	92	88	84	88	87	75	83	87	89	89	88	75	100	72	5	5	50	50	50	50	5	5	5
z	45	79	82	72	60	69	67	38	55	69	79	72	87	78	72	100	85	75	75	50	50	50	5	5	5
s	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	80	100	90	85	60	60	60	40	5	5
c	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	70	90	100	85	85	85	85	60	5	5
h	70	60	5	5	5	5	5	5	70	70	70	50	50	70	60	70	85	85	100	90	90	90	70	5	5
p	50	40	5	5	5	50	5	5	5	5	85	85	85	85	50	40	60	85	90	100	100	100	90	90	90
t	50	40	5	5	5	50	5	5	5	5	85	85	85	85	50	40	60	85	90	100	100	100	90	90	90
k	50	40	5	5	5	50	5	5	5	5	85	85	85	85	50	40	60	85	90	100	100	100	90	90	90
.	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	60	85	90	90	90	100	20	5
–	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	60	85	90	90	90	100	100	20
/	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	60	85	90	90	90	100	100	100
*	5	5	5	5	5	5	5	5	5	5	85	85	85	85	5	5	60	85	90	100	100	100	5	5	5

neme strings with the help of the similarity matrix.

The spectrum distribution of each phoneme was assumed a 20 dimensional normal distribution and the distance was defined by Bhattacharyya distance. Then, this distance was converted into the value from 0 to 100 by the linear transformation. But if a phoneme /i/ or /j/ was a voiceless consonant, the phoneme similarity S(i, j) was decided on the basis of the confusion matrix of phoneme Recognizer. Table 1 shows the phoneme similarity matrix. The row of table corresponds to a phoneme of the dictionary and the column a recognized phoneme.

For each word in the dictionary, the lexicon contains one description of that word as a string of phoneme symbols. Some phonemes in a word are often influenced by the context (so-called co-articulation). In consequence of this influence, these phonemes are often misrecognized as other phonemes or omitted. Therefore, we introduce a sub-phoneme /k/ in addition to a main-phoneme /J/ and denote this description in the dictionary by J/k (c), where c means the weight

Table 2. Examples of entries in the Word Dictionary.

Word	Symbol	Phoneme representation	Maximum duration	Minimum duration
ichi	1	i •/c(1.0) c i/c(1.0)	350 ms	100 ms
ni	2	n i	300	100
san	3	s a N	550	200
yon	4	y/g̃(0.9) o N	450	150
go	5	g o	300	100
roku	6	r/p(0.85) o •/k(0.95) k ů/*(1.0)	450	100
nana	7	n a/N(0.85) n/a(0.85) å/N(0.85)	550	200
hachi	8	h a/N(0.85) •/c(1.0) c i/c(1.0)	500	150
kyu	9	//c(0.95) k/c(0.95) y/u(0.95) u	500	200
rei	0	r/p(0.85) e i/e(0.95)	400	100

of the sub-phoneme /k/ ($0 \leqq c \leqq 1.0$). Table 2 shows the examples of lexicons in the word dictionary. The phoneme with a circle sybol (O) in the table indicates to be able to associate with one or two phonemes in the recognized phoneme string and the mark * indicates a pseudo phoneme (see Table 1). These descriptions for given words are automatically constituted by the constructing rules of the word dictionary.

Let I be the first candidate of the i-th recognized segment by Phoneme Recognizer, l the second candidate and p the reliability of recognization of the first candidate ($0 \leqq p \leqq 1.0$). Let J be the j-th main-phoneme of a given lexicon, k a subphoneme and c the weight of the sub-phoneme. Then, if that segment in a recognized string associates with the j-th phoneme of this lexicon, the similarity

is defined as the following equation:

$$S(I, l, p; J, k, c) = \max \begin{cases} S(I, J) \\ c \times S(I, k) \\ p \times S(I, J) + (1-p) \times (l, J) \\ p \times c \times S(I, k) + (1-p) \times c \times S(l, k) \end{cases}$$

We simply denote S(I, 1, p: J, k, c) by $S_0(i, j)$. Where we introduced the following restrictions with respect to the matching between a recognized phoneme string and a phoneme string of a lexicon.

1. Except for a phoneme marked with a circle (O), a vowel and a syllabic nasal in the lexicons are associated with phonemes of three or less in a recognized phoneme string.
2. A consonant in the lexicons is associated with phonemes of two or less.
3. Three successive phonemes in the lexicons are not associated with one phoneme in a recognized phoneme string.
4. Except for an elongated vowel, when the total duration of three successive phonemes in a recognized phoneme string is beyond 250 ms, a vowel in the lexicons need not be associated with the these phonemes.
5. When the duration of one phoneme is not beyond 100 ms, an elongated vowel in the lexicons need not be associated with only this phoneme.
6. If a word matching is performed beyond the range of the given duration by the lexicon by the matching score is reduced.

The evaluation score of matching is calculated by the average of the similarity for all phonemes in the lexicon. The likelihood for a given word is defined as the highest of all the scores. This can be obtained efficiently by the method of dynamic programming.

Let L(i, j) be the highest cumulative score up to the i-th recognized phoneme and the j-th phoneme of a given lexicon. When the j-th phoneme of the lexicon is a vowel, L(i, j) is calculated with the following equation:

$$L(i, j) = \max \begin{cases} L_1(i, j) = L^*(i, j-1) + S_0(i, j) \\ L_2(i, j) = L(i-1, j-1) + S_0(i, j) \\ L_3(i, j) = L^*(i-1, j-1) + [S_0(i-1, j) + S_0(i, j)]/2 \\ L_4(i, j) = L(i-2, j-1) + [S_0(i-1, j) + S_0(i, j)]/2 \\ L_5(i, j) = L^*(i-2, j-1) + [S_0(i-2, j) + S_0(i-1, j) + S_0(i, j)]/3 \\ L_6(i, j) = L(i-3, j-1) + [S_0(i-2, j) + S_0(i-1, j) + S_0(i, j)]/3 \end{cases}$$

where $L^*(i, j-1) = \max\{L_1(i, j-1), L_2(i, j-1), L_3(i, j-1), L_4(i, j-1), L_5(i, j-1)\}$ (see restriction 3).

In the case of a consonant, L(i, j) is max $[L_1, L_2, L_3, L_4]$. If the length of the lexicon is j_0, the likelihood is calculated as $\max_i L(i, j_0)/j_0$. To avoid an identification error, best i_1 and 2nd best i_2 are calculated for all i. In this way, Word

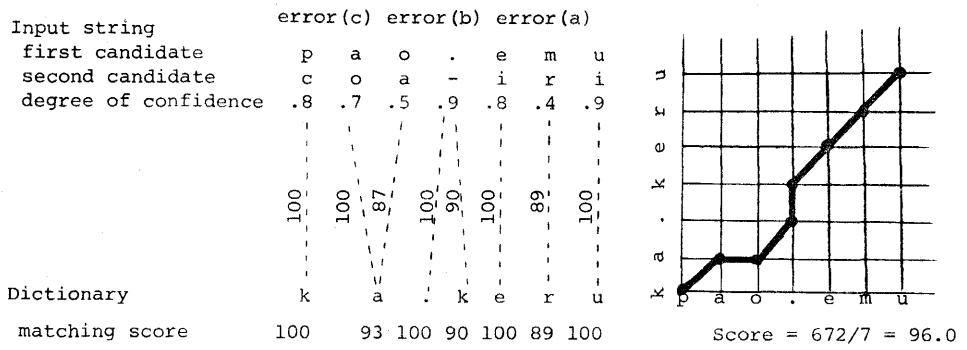

Fig. 4. Graphic representation ot word matching and matching score.

Identifier generates two locations in a recognized string for a given word. Fig. 4 shows an example of a matching and the route of the word "kakeru" (times).

We made an experiment to recognize isolated spoken digits by this method. Japanese digits have an average of 1.7 syllables. LITHAN correctly recognized 97% for 1500 words of 20 male speakers and 98% for 500 words of 10 adaptive speakers.[5]

So far the way of the identification of a word has been explained. Next that of a sequence of words is explained. In the case of continuous speech, a recognized phoneme string which corresponds to the word identified lately often consists of some segments. So, we combine this phoneme with the next identified word, and this new phoneme string is identified in a recognized phoneme string. In this case because there may exist a pause between the two successive words, we introduce a sub-phoneme '/' for the last phoneme. An example shown in

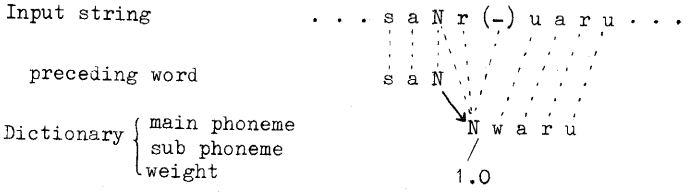

Fig. 5. Sequential matching method.
( ) means optional phoneme

Figure 5 indicates that the preceding word is "san" (3) and that the word to be identified is "waru" (divide). When the word "waru" is identified, the last phoneme of "san" and "waru" are connected. This is called sequential matching method.

If pseudo phoneme /y_{psd}/ is combined with the first phoneme of the word to be identified and this new phoneme string is identified in all the range of an input string, then we can obtain the likelihood of this word in arbitrary portion of a input string. Where the similarity between /y_{psd}/ and a recognized phoneme is

```
input                  kesaN ki cyuono zikitepu so  ci saNbaNkarakesaN kigazo e d e  tayoNoro doseyo
segment number         0.....1.....2.....3.....4.....5.....6.....7.....0.....1.....
first candidate        /pucau.pi.cuuoro-puciges.csou.ci.caobaopao-pucaN.pidazouu.bgeu.cabogoroogusee/
second candidate       -cisegpcuphiNuuu.hihuducppcuopsbpsoXrorcor/cisXuphuGoduee-ubuXphonNuubeudocug-
reliability (×100)     61  6  11612642    13 24916 29229 3 73 2  11 1313 31 14 342 1 9471 22113 243 6
                       06620006624260208060680980609289804840000208226648606200044469844664604444440
```

lexicon

main pho.	/	k	e	i	s	a	N	.	k	i		s	o	.	c	i	
sub pho.		k		e					k	*					c	c	
										1					1	1	
weight (×100)		9		9					9	0					0	0	
		5		5					5	0					0	0	

keisanki																	
score	94	94	94	91	89	90	91	95	95	94	89						
head	0	0	0	0	0	0	52	52	52	52	52						
tail	6	7	10	11	12	13	60	61	62	63	64						
sochi																	
score	90	91	93	89	90	92	89	89	97	98	94	90	91	92	93	91	
head	1	3	3	3	7	21	21	23	32	32	32	36	53	55	55	55	
tail	3	7	10	11	13	24	25	31	36	37	40	41	55	61	62	63	

Fig. 6. Detection of "keisanki" and "sochi" by direct matching method.

0. This is called the direct matching method as opposed to the sequential matching method and is used for the detection of the key words in utterances. Figure 6 shows this example. Utterance is "Keisanki cyuono zikidisuku sochi sanban kara keisanki gazoe deta yono rodoseyo". (Load the 4th datum from the 3rd magnetic disk device of central computer to imageprocessing computer.) There are three key words, i.e. "dengen" (power souce), "keisanki" (computer) and "sochi" (device). A key word "dengen" was not detected in score above 89 in this utterance. The new lexicon of "keisanki" is "y_{psd}/keisaN.ki". Overlapping locations are remained only one location where the matching score is the highest. In the case of "keisanki", two locations, [0,6] and [52,61] are detected finally (the actual locations are [0,7] and [53,62]).

4. Word Predictor

The syntactic rules are represented as context free grammar without including recursive rules and are given to this sysem as a given knowledge. Of course, this grammar can generate all possible sentences for the task. In addition, the grammar has the additional information such as phonological rules. Each word is classified into several classes syntactically and semantically, and a word may belong to some classes. This class is represented by a nonterminal symbol and called a word class for convenience. Therefore, a word class generates all words belonging to this word class.

In Japanese, the predicative part is situated to the last portion of a sentence and the role of predicates are very important on syntactic information. Therefore, LITHAN treats the predicates in particular. A partial sentence which

Table 3. Relation between predicate and sentence structure.

Predicate	Sentence structure	Rewriting rule	Key Word					
			Dengen		Keisanki		Sochi	
			mini	max	mini	max	mini	max
Aiteiruka	<P1>	<R1>→itsu <R2>→dorega	0	0	1	1	0	1
Tsukatteiruka	<P1>	<R1>→darega <R2>→doreo	0	0	1	1	0	1
Hashitteiruka	<P2>	——	0	0	1	1	0	0
Kakeyo	<P9>	——	0	0	1	1	1	1

is obtained by eliminating only a predicate from a sentence is called a sentence structure. Table 3 shows the examples of the relation between a predicate and a sentence structure. When different predicates have the same sentence structure except partial words, we assign the same sentence structure to these predicates. These different parts are processed by the rewriting rules corresponding to the predicate (see Table 3). By this method, we can reduce the sentence structures and can avoid making complicated grammar.

Furthermore, LITHAN permits the rewriting rules of AB→AC type, that is, context sensitive, where A and B are either nonterminal or terminal symbols. The grammar of the system becomes a more flexible one by this description. Of

<P1>::=<Q2><Q1> | <D1>$_3$ sochi<Q7>
<P2>::=<Q2>de$_4$ zyobu$_5$ wa<WW>
<P9>::=<Q2>no$_4$<D8>$_{1,2,3}$ sochi<WS>$_5$ ban$_4$<WJ>$_4$<Q6><WS>$_5$ ban$_4$ O$_4$
<Q1>::=<Q7> | no$_4$<T8>
<Q2>::=keisanki<WK>$_{1,5}$
<Q6>::=<WI><D2>$_2$ | <D6>$_2$
<Q7>::=wa$_4$<U1>
<U1>::=<R1> | ε
<U7>::=ikutsu | <R2> | ε
<T6>::=<WS>$_5$ ban$_4$
<T7>::=<D1> | <D3> | <D4> | <D5> | <D7>
<T8>::=<T7>$_{1,3}$ sochi<Q7> | <D8>$_{1,3}$ sochi<TB>
<TB>::=<T6>wa$_4$ | wa$_4$<U7>
<WW>::=ikutsu | dorega
<WS>::=ichi | ni | san | yon | go | roku | nana | hachi | kyu | rei
<WJ>::=ni | e
<WK>::=kokan | cyuo | hanyo | onsei | gazo | gengo
<WI>::=aoi | shiroi | kiiroi | akai
<D1>::=zahyonyuryoku | onseinyuryoku | onseisyutsuryoku
<D2>::=zikitepu | kasettozikitepu
<D3>::=kamitepuyomitori | kadoyomitori
<D4>::=taipuraita | kosokuinzi
<D5>::=mozihyozi | gazohyozi | shikisaihyozi
<D6>::=zikidoram | zikidisuku
<D7>::=kamitepusenko | kadosenko | nizigenhyozi | gazonyuryoku
<D8>::=<D2> | <D6>

Fig. 7. Examples of grammar.

course, Word Predictor memorizes the route in the grammar for each word string (A word string is called a partial sentence).

LITHAN has the restrictions with respect to the grammar as follows:

1. to be context free grammar without including recursive rules.
2. to be an unambiguous grammar.
3. can be applied unconditionally whenever the rewriting rule of AB→AC type can be applied.

Some examples of the grammar are shown in Fig. 7. The grammar consists of the syntactic rules and the additional information. We explain this additional information in the following.

Table 4. Connective relation between a computer and I/0 devices or computers.

Computer	I/0 device
	computer
Kokan	taipuraita (TTY), zikitepu (MT), kamitepuyomitori (PTR), mozihyozi (CRT)
	cyuo, hanyo, onsei, gazo, gengo, gaisen
Cyuo	taipuraita (TTY), kosokuinzi (LP), zikitepu (MT), zikidisuku (DSK), kamitepuyomitori (PTR)
	kokan, onsei, gazo

TTY: Tele type writer, MT: Magnetic tape, PTR: Paper tape reader,
LP: Line printer, CRT: Cathode ray tube, DSK: Magnetic disk

1. subscript 1 Table 4 indicates the name of computers and I/O devices which have connection with each other. If Word Predictor has predicted a symbol (nonterminal or terminal) with the subscript 1, it predicts only words which are consisted with the content of Table 4. In Figure 7, for example, when the word class $<D8>_1$ has been predicted in a partial sentence "Keisanki cyuono...", Word Predictor predicts only "zikitepu" and "zikidisuku" by the row "cyuo" of Table 4.

2. subscript 2 In the same manner as the subscript 1, in some sentence, two symbols marked with a subscript 2. The latter symbol with the subscript 2 (in general a word class) is replaced by the former symbol with it unconditionally. This means that the latter symbol can be replaced by a demonstrative pronoun such as "sore" (it). For example of Figure 7, when $<D8>_2$ is "zikidisuku", since $<D2>$ derived from $<Q6>\rightarrow<WI><D2>_2$ does not include "zikidisuku", $<D2>$ is rejected and Word Predictor predicts only $<D6>$ derived from $<Q6>\rightarrow<D6>_2$, and $<D6>$ is replaced by "zikidisuku".

3. subscript 3 When Word Identifer identifies a word with the subscript 3, it is necessary for a key word to have been found just immediately following place of the location to be identified in a recognized phoneme string. In Figure 7, for example, when Word Predictor predicts $<D8>_2$, it is necessary for a word

"sochi" to have been found.

4. subscript 4 In Japanese, there does not exist a pause between a noun and a preposition or a number and a unit of the number. The subscript 4 indicates that there does not exist a pause in front of a word with this subscript. Therefore, in the sequential matching method, the sub-phoneme '/' of last phoneme of the preceding word is not taken into consideration.

5. subscript 5 In general, the longer the length of a word is, the better the reliability of a matching score becomes. The word with the subscript 5 is connected with the one the just immediately following it and then identified. Because the computation time expresses a linear function of the length of the word and the number of predicted words, the subscript 5 is affixed on only the word that is followed by one word obviously. In Figure 7, if the word "zyobu" (job) is predicted, Parsing Director combines "zyobu" with the following word "wa" (a kind of preposition), and Word Identifier identifies "zyobuwa".

6. subscript 6 The subscript 6 is affixed on the word B so that we may apply the rewriting rule of AB→AC type. If C is empty, B is rejected, and if A is empty, of course, B is replaced by C unconditionally.

7. Let X be the last phoneme of the preceding word and Y be the first phoneme of the word to be identified. For avoiding the effect of the context between words, we introduce the rewriting rule of XY-xy type. This rule means that the sub-phonemes of X and Y are replaced by x and y, respectively.

8. The representation of "A kara B made" (from A to B) usually represents the relation of time order between A and B. In our task, this representation appears for user reservation of computer resources such as "gozen 8: 30 kara gogo 3: 30 made" (from 8: 30 AM to 3: 30 PM). Word Predictor predicts words utilizing the fact that the ending time comes after the starting time.

5. Parsing Director

The words which have been predicted by Word Predictor are identified by Word Identifier and their likelihood is calculated. But it often happens that a misidentified word has higher likelihood than the word actually uttered because the recognized phoneme string is noisy or errorful. Therefore, Word Identifier should accept many probable words than a single word only. On the other hand, since a word boundary in continuous speech is generally not clear, a predicted word should be identified at the various places in a recognized phoneme string. In short, when the position in a recognized phoneme string corresponding to the last phoneme of a partial sentence differs from that of the other partial sentences, it is necessary that we must consider those two partial sentences concurrently. As the result of parsing, we can get the tree which consists of strings which do not overlap but connect with words each other. Parsing Director calculates the likelihood of these word strings and Word Predictor predicts words at the unknown

portion of same word strings (i.e. partial sentences) with higher likelihood. After we repeat this process (called one round), the word strings with higher likelihood become longer gradually. After all, we can obtain complete sentences. We decide that the final recognition result is the word string with the highest likelihood in all probable sentences.[6]

Since the predicate plays the important role in regard of syntax and also the position of the predicate is the last of a sentence in Japanese, we identify the predicates in the latter portion of a recognized phoneme string. If a predicate was identified, the sentence structure is dicided by the relation of Table 3 and Parsing Director performs a left-to-right parsing while controlling Word Predictor and Word Identifier. In this process, because the number of the branches of a word tree becomes enormous, some pruning method is necessary. We apply the following pruning method.

Pruning method. Let α be the number of nodes expanded at the same time. If the number of the nodes which have been generated by an expanded node exceeds β, only the best β nodes are remained. If the total number of new word strings and nonexpanded word strings exceeds γ, only the best γ word strings are remained.

These consecutive process is called one round. If no word string has better likelihood than that of the next best predicate, the sentence structure corresponding to this predicate is newly expanded. Parsing Director repeats this process until a reliable complete sentence is generated. Although the larger α, β and γ are, the better the recognition result becomes, the tree search time becomes longer inversely. Therefore, we wish to decide the optimum thresholds of α, β and γ. The preliminary results of speech recognition of arithmetic expressions (see Appendix) showed, the case of $\alpha=5$, $\beta=5$ and $\gamma=15$ was almost saturated

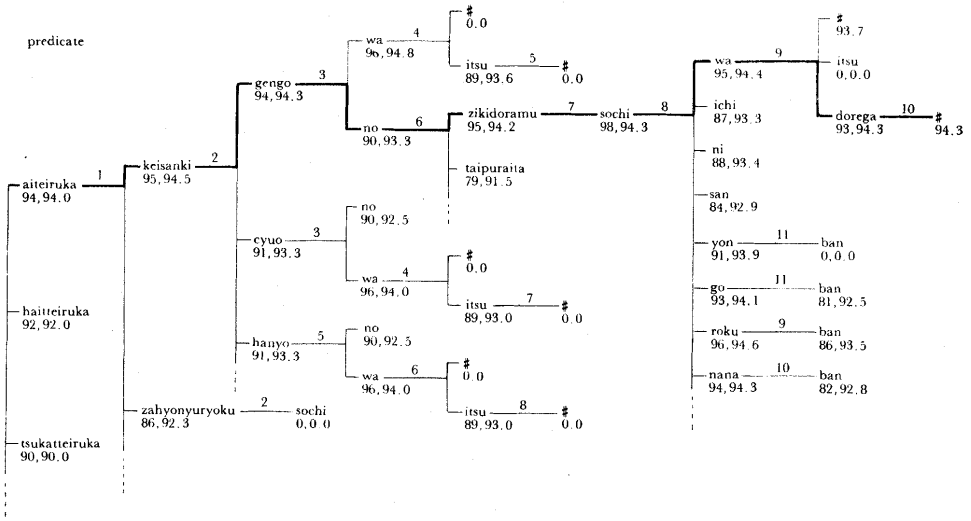

Fig. 8. An example of parsing.

and also since the case of $\alpha=2$, $\beta=5$ and $\gamma=10$ led to good result, we used these thresholds in the expriments stated below.

Figure 8 shows an example of parsing. The utterance is "Keisanki gengono zikidoramu sochiwa dorega aiteiruka?", which means "Which device is ready among the magnetic drum devices of language processing computer?" The thick line indicates a correct word string. The left number under a word indicates the likelihood of this word, the right number the likelihood of a word string up to this word and the upper number of a branch the order of an expanded round. The symbol "#" denotes the end of a complete sentence. Firstly, Parsing Director tries to detect three key words, which are "dengen" (power source), "keisanki" (computer) and "sochi" (device), in the recognized phoneme string. We can obtain the number of key words, the locations and the likelihood. Then, we can restrict or estimate some predicates by these results and the contents of the columns of key words in Table 3, which indicate the minimal and maximal number of each key word to be included in a given sentence structure. Next, Word Identifier identifies the predicates at the latter portion of the recognized phoneme string. In Figure 8, as the likelihood of "aiteiruka.' (is ready...?) was the highest in one of all predicates, the sentence structure corresponding to this predicate was parsed. In the midst of parsing, "Keisanki gengowa aiteiruka?" (Is language processing computer ready?) and "Keisanki cyuowa itsu aiteiruka?" (When is central computer ready?) etc. were generated. Parsing Director rejected these complete sentences, however, because there was still much of the spoken input unaccounted for.

6. Results and Discussions

The similarity matrix and reference patterns of vowels, semivowels and voiced consonants were calculated from [vowel$_1$-semivowel-vowel$_2$] and [vowel$_1$-voiced consonant-vowel$_2$] contexts which were included in 2450 words for 10 male speakers.

The current implementation of the LITHAN system was tested on a total of 200 sentences containing 1983 words, spoken by ten male speakers at a normal speed and using our task (described briefly in the introduction). A sentence consists of 4 to 22 words, 10 words on an average. Ten sentences are uttered commonly by all speakers. The LITHAN system correctly recognized 128 of

Table 5. System performance.

Task	No. of vocabulary	No. of speaker	No. of sentence	No. of word	Recognition rate	
					Sentence	Word
Digit	10	20	——	1500	——	97%
Arithemetic Expression	24	5	80	560	71%	94
Computer Network	101	10	200	1983	64	93

200 utterances or 64% and also recognized 38 utterances having differed from the actual utterances by one word. Out of these 38 words, 17 words were digits. When the pragmatic information, that is, the subscript 1 was not used, 58% of the utterances were recognized correctly.

The correct rate of the sentence recognition was very influenced by the speakers and the speed of an utterance. For example, 18 of 20 utterances were recognized correctly for the speaker MG and 6 of 20 utterances for the speaker FZ.

On the other hand, about 93% of the words in output sentences were recognized correctly. The correct rate of the digits was about 80% and much worse than that of isolated spoken digits[5]. This result suggests that continuous speech recognition is very difficult.

Although the number of the predicted words at a time was 20 words in case of many ones except the predicates (of course, became 21 words in case of predicates), it was almost under 7 words because of utilizing various linguistic information.

All of the 352 key words were correctly detected except for two but the number of detected words was about two times that of key words.

The LITHAN system allows new tasks to be implemented with minimal effort. For example, only one man-day was required to implement the recognition of arithmetic expressions.

Table 6. Decomposition of the LITHAN system.

Level	Acoustic	Parametric	Phonemic	Lexical	Sentence
Subject	speech	feature parameter	segment	phoneme string	word string
Object	feature parameter	segment	phoneme	word	sentence
Operation	spectrum analysis	heurisic	Bayes rule	dynamic programming	tree search
Operational order	left-right	left-right	left-right	left-right right-left	left-right right-left
Procedure	filter-bank	feed back	learning	bottom-up	top-down

The composition of the LITHAN system is shown in Table 6. From this table and Fig. 1, it is seen that there does not exist the interaction between the lower levels and higher levels in the current system. From the view point of the system performance (ie., computing time and recognition rate), we think that this interaction is necessary.

We also think that the most important area for future research is to develop techniques such as the normalization of the effect of the variation of phoneme patterns among speakers and in contexts. Furthermore, it is necessary to introduce prosodic information.

Acknowledgement

The authors wish to thank Assistant Professor S. Sugita and Dr. T. Kanade for their helpful advice concerning the research for and writing of this paper. They also wish to thank Mr. N. Yoshitani, K. Maegawa and T. Ukita for co-operation and various conveniences.

References

1) T. Sakai and S. Doshita, "The automatic speech recognition system for conversational sound" IEEE Trans. vol. EC-12, 1963.
2) Newell, et al., "Speech Understanding Systems: Final Report of a Study Group" North-Holland, 1973.
3) Proceedings IEEE Symposium on Speech Recognition, Carnegie-Mellon University, April, 1974.
4) T. Sakai, et al., "Segmentation and phoneme recognition of conversational speech" 1973 Joint Convention Record of Four Institutes of Electircal Engineers, Japan.
5) T. Sakai and S. Nakagawa, "A Word Identification Method in Continuous Speech" Record of Joint Convention of the Acous. Soc. of Japan, May, 1975.
6) S. Nakagawa and T. Sakai, "Utilization of word string's information" 1974 Joint Convention Record of Four Institute of Electrical Engineers, Japan.

(Aug. 31, 1975, received)

Appendix

Speech Recognition of Arithmetic Expression

<UT>::=<EX> | <EX>=
<EX>::=<NM><EF> | <EM><OP1><NM>
<EF>::=<OP1><EY> | <OP2><NM><EG>
<EG>::=<OP><NM> | ε
<EY>::=<EM> | <NM><EG>
<EM>::=LK<NM><OP2><NM>RK
<NM>::=<IN><IT>
<IT>::=.<DQ> | ε
<IN>::=<SN><D5> | <D5>
<D5>::=<DG><DS> | S<D4> | <DF> | 0
<DS>::=S<D4> | <D3>
<D4>::=<DG><D3> | <DF> | ε
<DF>::=F<D2> | Z<D1> | 1
<D3>::=F<D2> | Z<D1> | ε
<D2>::=<DG><DZ> | Z<D1> | 1 | ε
<DZ>::=Z<DC> | ε
<D1>::=<DC> | ε
<DQ>::=<DG> | 0 | 1
<DC>::=<DG> | 1
<DG>::=2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
<SN>::=P | M
<OP>::=+ | − | × | /
<OPI>::=× | /
<OP2>::=+ | −

Fig. A.1. Generative grammar of Arithemetic Expression
ε means empty, S: sen (thousand), F: hyaku (hundred), Z: zyu (ten), P: purasu (plus), M: mainasu (minus), L: hidari (left), R: migi (right), K: kakko (parenthesis)

Figure A.1 shows the generative grammar of the arithmetic expressions by the description of the Backus Normal Form. The vocabulary size is 24 and the lexicons consist of 2.1 syllables on the average. The number was restricted less than 10,000 with sign or not and was permitted to one decimal place. And also this grammar restricted within one parenthesis and two operators in one sentence. For example, this grammar can generate the sentence such as "$1.2\times(345-6789.0)=$".

Table A.1. Recognition results of Arithmetic Expressions.

α	β	γ	Search time	Recognition rate	Notes
1	1	1	1.0	30.0%	No back tracking
2	2	2	2.0	47.0	
3	3	3	3.0	56.0	
4	4	4	4.0	58.0	
5	5	5	5.0	60.0	Standard
6	5	6	6.0	61.0	
1	5	10	1.9	52.0	(Nearly) Best first method
2	5	10	2.6	58.0	
3	5	12	3.3	60.0	
5	5	15	5.1	62.0	(Rough) Breadth first method
5	5	5	12.0	20.0	No syntactic information
5	5	5	5.0	45.0	Utilizing other similarity matrix*
5	5	5	5.0	53.0	Neglect of second best phoneme
5	5	5	5.0	40.0	Neglect of confident degree of phoneme
5	5	5	5.0	54.0	Neglect of a preceding word information
5	5	5	5.0	58.0	Neglect of duration times of words
5	5	5	5.0	64.0	Two identified locations
5	5	15	5.1	71.0	Final result (Utilizing subscript 5)

* Drived from distinctive features.

The experiments of speech recognition of arithmetic expressions were tested on a total of 80 utterances containing 560 words, spoken by five male speakers at a normal speed. The results are shown in Table A.1 and A.2. The column of the search time in Table A.1 indicates the ratio of the number of identified (or predicted) words in each to that in the case $\alpha=1$, $\beta=1$ and $\gamma=1$. The number of predicted words at a time was about 10 words on the average. The method corresponding to the final result used the subscript 5 (i.e., L_5 and R_5) and others.

Table A.2. Detailed result of speech recognition of Arithmetic Expression.

Speaker	SK	OT	MG	NK	YS	Total
Sentence	94 %	59 %	66 %	69 %	66 %	71 %
Word	99.1	92.4	94.2	93.1	91.5	94.0
Vowel, /N/	88.1	82.1	92.7	85.6	80.3	85.9
Semi-vowel	28.6	28.6	14.3	42.9	35.7	30.0
Voiced consonant	33.7	31.7	39.6	37.5	28.8	34.3
Unvoiced consonant	90.2	84.1	72.0	74.6	83.5	81.0

Vowel /a/,/i/,/u/,/e//o/ Semi-vowel /y/,/w/
Voiced consonant /m/,/n/,/ğ/,/b/,/d/,/g/,/r//,z/
Unvoiced consonant /s/,/c/,/h/,/p,t,k/ Syllabic nasal /N/

The LITHAN system recognized 71% of sentences and about 94% of words correctly. And also this system recognized 18% of the sentences having differed from the actual sentence by one word. The detailed result of the experiment is shown in Table A.2. The result of phoneme recognition shows the correct rate of the first candidate of Phoneme Recognizer.