IEICE TRANS. INF. & SYST., VOL. E76-D, NO. 1 JANUARY 1993

PAPER Special Issue on Speech and Discourse Processing in Dialogue Systems

A Spoken Dialog System with Verification and Clarification Queries

Mikio YAMAMOTO†, *Member,* Satoshi KOBAYASHI†, Yuji MORIYA†, *Nonmembers and* Seiichi NAKAGAWA†, *Member*

SUMMARY We studied the manner of clarification and verification in real dialogs and developed a spoken dialog system that can cope with the disambiguation of meanings of user input utterances. We analyzed content, query types and responses of human clarification queries. In human-human communications, ten percent of all sentences are concerned with meaning clarification. Therefore, in human-machine communications, we believe it is important that the machine verifies ambiguities occurring in dialog processing. We propose an architecture for a dialog system with this capability. Also, we have investigated the source of ambiguities in dialog processing and methods of dialog clarification for each part of the dialog system.

key words: natural language processing, speech recognition, dialog system, verification, clarification

1. Introduction

Recent research in dialog systems has shown a significant improvement over earlier question-answering systems. Concern is focussed on a mechanism for more natural conversation.[1] The ultimate system should have powerful inference ability and knowledge, allowing it to disambiguate in the process of analyzing the structure and meaning of sentences, resolve ellipses, and infer pronoun references and user intention. However, natural language conversation with machines still cannot guarantee reliable wide-ranging conversation. If natural conversation is the aim, many problems remain to be solved.

As for the inference of concealed user intentions in an utterance, a correct inference benefits the user. In contrast, a wrong inference of intention only confuses the user. In the worst case, misunderstanding occurs.

As for ambiguity at the context level, disambiguation may be difficult even for humans. In daily conversation, we often ask questions for verification or clarification concerning the meaning of an utterance. Since the ability of a machine to disambiguate is inferior to that of man at the current state, a dialog system should actively interact with the human user for help in cases of ambiguities. Moreover, we believe a system that can carry on a reliable conversation will become important from the viewpoint of the practical use of natural language dialog systems.[2],[3]

Manuscript received June 22, 1992.
Manuscript revised August 26, 1992.

† The authors are with the Faculty of Engineering, Toyohashi University of Technology, Toyohashi-shi, 441 Japan.

2. Analysis of Verification and Clarification within Human Dialog

In this section, we analyze the content, types, and responses of the verification and *clarification* query in transcripts of human-human spoken dialog.

2. 1 Transcripts of Human-Human Spoken Dialog

We analyzed nineteen transcripts (1769 Japanese sentences) in the continuous speech corpus for research of the Acoustical Society of Japan. They were collected from telephone conversations between Japanese. They are information seeking dialogs such as at a travel bureau or of other such consultations.

2. 2 Analysis of the Contents of Verification and Clarification Queries

We analyze verification and clarification queries for the ambiguous sentences. The following sentences are categorized into verification and clarification queries (the second sentence of the examples). We define that verification queries ask what another person explicitly stated and clarification queries ask what another person implicitly says or does not say.

In the examples, Q and A represent a questioner and an answerer, respectively. Note that in example (5), since the answerer suddenly speaks about a facsimile not related to the previous context, the questioner cannot understand the relationship between the facsimile and the context. The questioner represents his lack of understanding with "Pardon ?".

Clarification queries

(1) information seeking to answer questions

Q〉 ikura kakarimasuka ?

(How much does it take ?)

A〉 ninzū-wa nan-nin desuka ?

(For how many persons ?)

(2) refinement of meaning

Q> kisha-wo tsukaitain desukedo.

(I'd like to take a train.)

A> soreja shinkansen desune.

(You mean the shinkansen superexpress, right?)

(3) deciding the level of answer

Q> michijun-wo oshiete hoshiin desukedomo.

(Could you tell me the way?)

A> ee, Toukatta-onsen-wa oideni natta koto gozaimasuka?

(Have you ever been to Toukatta-onsen?)

(4) Meaning of word

A> Ryūou-to iu tokoro desu.

(A place called Ryūou.)

Q> "Ryūou"-wa chimei desuka?

(Is "Ryūou" the name of a place?)

(5) representation of not understanding

A> fakkusu-de okurimashouka?

(May I send it by facsimile?)

Q> ha?

(Pardon?)

verification queries

(6) meaning of sentence

A> Kyou-fū-no kaiseki-ryouri-ga yūmeina tokoro-ga Motohakone-to iu tokoro-ni…

(There is a restaurant famous for Kyoto style cuisine lunches in Motohakone.)

Q> Hakone-de Kyou-fū desuka?

(Kyoto style in Hakone?)

(7) whole sentence or sentence fragment

Q> dekireba wafū-ga iindesuga.

(If possible, I prefer Japanese style.)

A> wafū desuka?

(Japanese style?)

Q> hai.

(Yes.)

(8) sound of sentence fragment

A> hoteru "Nōsu".

(Hotel "North")

Q> Nōsu"?

("North"?)

A> ee-soudesu.

(Yes.)

We categorize queries into three types: "yes/no", "what" and "which" types. For "yes/no" queries, one can simply reply "yes" or "no". For "what" queries, one should answer with a value of some object. For "which" queries, one makes a decision among the alternatives. The result of this analysis is shown in Table 1.

Since information seeking queries to answer question (1) are not much concerned with clarification, we can regard them as ordinary questions. Excluding (1), there are 94 verification and clarification queries. However, since answeres in dialog accompany questions, at least twice that, or 188 sentences concern verification and clarification. That is, ten percent of all sentences are related to verification and clarification. This result is obtained from human-human conversations. The machine dialog system may ask more questions. This is particularly important to a spoken dialog system because of the ambiguities in the speech recognition process.

There are many verifications and clarifications of sentence meanings and of partial or whole sentences. These show that it is important to understand exactly the meaning or intention of a user utterance and that even a human cannot always understand these immediately. Clearly these types of queries are important for a computer dialog system using current technology.

2.3 Analysis of Answers to Queries

When one asks general verification questions such as "Pardon?", in many cases the other person responds with a more complex and longer answer than the previous ambiguous statement. In general, it is difficult for a machine to understand a complex and long utterance. To facilitate machine understanding, the

Table 1 Categorization of verification and clarification queries.

Type / Contents	yes/no	what	which	total
(1)	10	18	6	34
(2)	24	0	0	24
(3)	6	0	0	6
(4)	2	0	0	2
(5)	0	1	0	1
(6)	16	0	0	16
(7)	37	0	0	37
(8)	6	2	0	8
total	101	21	6	128

machine should generate a query to which another person responds with a short and simple answer. In this section, we analyze the type of query with which the machine can obtain a short and simple answer.

There are characteristic patterns for each type of query. We show the result of analyzing responses into query types in Table 2. "Yes" or "no" of the "yes/no" type means that the first word of the answer is "yes" or "no". "Complex" means that the answer is a complex sentence where "yes" or "no" can be understood by inference. "Repeat" means that the answer is a repeat of part of the question. "Nothing" means no answer. "Respond" of "which" type query means that another person can answer and "no respond" means that s/he cannot answer. "Added info." and "no added info." mean that the answer includes and does not include information in addition to the literal answer, respectively. Answers of "yes", "no" and "repeat" can be understood by the machine, and in particular, the "no added info." type is very easily understood, because the type or pattern of answer is restricted. Such answers make up 70 percent of all answers of the "yes/no" and "which" queries. Therefore, questioning by "yes/no" or "which" queries should promote machine understanding.

The results in the previous and in this section do not apply to all kinds of dialog. However, the results are, we think, general for the information seeking dialog that is an important portion of the kinds of dialog from the viewpoint of the man-machine interface.

In the following sections, we describe a spoken dialog system that can generate a verification or clarification query based on an extension of the analysis of human-human communications to machine-human communications.

3. Spoken Dialog System

3. 1 Outline

In this section, we describe our spoken dialog system which simulates a fictitious Mt. Fuji travel bureau. The reason for selecting this domain is that there are many studies for natural language information retrieving systems. Also, a system for sightseeing information has value in real world applications.

Table 2 Categorization of responses.

		no added info.	added info.
yes/no	yes	66	12
	no	0	5
	complex	0	10
	repeat	6	0
	nothing	(2)	0
wh	respond	3	1
	no respond	0	2

Figure 1 shows the composition of the system. The speech recognition part outputs a phoneme (or word) string of an utterance as input to the understanding part. The understanding part is the extension of our question-answering system developed earlier. This part analyzes the morphological and bunsetsu (phrase) structure and then the dependency structure between phrases.

The dependency structure is transformed into a semantic network. The semantic network is sent to the context processing part that determines pronoun references and resolves ellipses based on the context. The context is the set of previous semantic networks of input sentences that are stored in the context stack, and a temporal semantic network for predicting the next user utterance.

With the use of the semantic network completed according to context and dialog rules, the utterance generating part generates the next utterance of the system. In the speech generation part, the generated utterance is transformed into real speech.

When an ambiguity or uncertainty arises in the above processing, the system pushes the status at that point into the system stack and activates verification or clarification dialogs to disambiguate it. At the end of the dialog, the system pops the stored status from the system stack and resumes processing from the point of ambiguity occurrence. Thus the system recursively calls the entire dialog system to verify or clarify ambiguous dialogs.

3. 2 Speech Recognition

We use the connected word recognition system developed in our laboratory.[4] The recognition system is based on a word spotting technique and a frame synchronized beam search with the constraints of context-free grammar.

Although the speech recognition part requires a grammar by which it removes illegal sentences as candidates from the recognition result, the understanding part also requires a grammar to find the structure of a legal input sentence. Therefore, our dialog system has two different grammars for the different aims. In addition, the separated grammars are useful for independent development of the parts of the system. Figure 2 shows a context-free grammar for the recognition system. The position of each symbol in a production rule is represented by an ordered number. An ordered number is calculated by adding the column and row numbers. Storing the history of rule applications as a string of numbers, the system can predict the next word efficiently by the top-down procedure.

In the recognition process, the prediction of successive words by sentence hypotheses and word spotting are performed in parallel from the left-to-right. For each word that has been predicted, the word

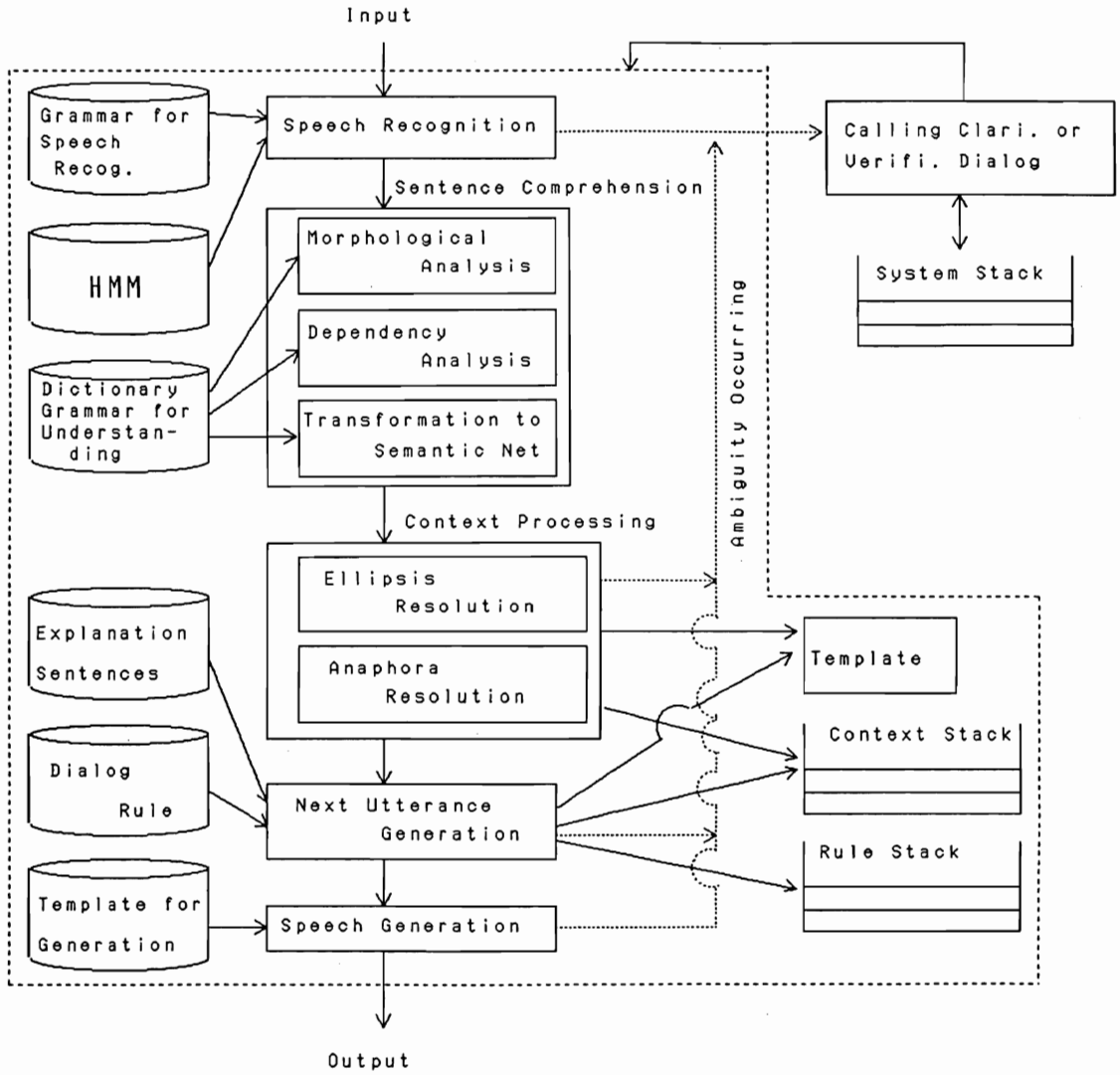

Fig. 1 Block diagram of the system.

spotting score and its beginning position are obtained using the Viterbi algorithm. We constructed the word models by concatenating syllable-based HMMs. Our algorithm incorporates word-level processing at the sentence level without a word lattice. Like word spotting, it uses only the highest of the sentence hypotheses scores to predict the same word as the initial score of the word. Since the search in our algorithm is synchronized framewise, it can easily employ the pruning method for search trees. We have experimentally shown that our approximate algorithm achieves sentence accuracy comparable to optimal algorithms.

The syntax of the task sentences related to Mt. Fuji sightseeing consultation is described by the context-free grammar that is represented by rewriting 239 rules without directly rewriting rules from word classes to terminal symbols. The vocabulary size is 426 words and the task perplexity is 27. The experimental

```
      0          1       2      3   ...
  8   @S   -->   @NP     @VP
 16   @S   -->   @NP     *AUX   @VP
 24   @S   -->   *AUX    @VP
 32   @NP  -->   *DET    @NP2
 40   @NP  -->   @NP2
 48   @NP2 -->   *ADJ    @NP2
```

Fig. 2 An example of CFG for speech recognition part.

result shows the sentence recognition rate of about 70 percent in the speaker adaptation mode.

3. 3 Sentence Comprehension

For analyzing morphology and Japanese bunsetsu (phrase), possible concatenations of words are represented by a finite state automaton. Kakari-Uke relationships (dependency relationships) of the Japanese language are analyzed by Kakari-Uke rules (dependency grammar). As the result of analysis, the

```
(node-name  class  (slot1  node-name1)
                   (slot2  node-name2)
                      ⋮                )
```

Fig. 3 Syntax of semantic network.

dependency structure is transformed into a semantic network based on the case structure of verbs. We employ selectional restriction to determine whether a phrase can or cannot modify another phrase. Each noun has three kinds of semantic features that are determined by three different viewpoints. A detailed discussion has appeared in Ref.(2).

A characteristic of the Japanese spoken language, the dropping of postpositions (Japanese joshi), causes analytical difficulty when using Kakari-Uke rules. We have investigated this problem and have shown that almost all dropped postpositions have a required-case role and we can resolve dropped postpositions using some heuristics and simple semantics with a success rate of 90 percent.[5]

The semantic network form is shown in Fig. 3. Node-name is a unique name to represent a node. Class represents a concept of the node. The node is related to node-name$_i$ by relation slot$_i$. The set of possible slots includes the following: "form", "wh-kind" and "target" that represent the sentence form, such as declarative sentence; "yes/no" or "what" questions, including the type of interrogative pronoun; and queried slot of "what" or "which" question, respectively. The following is the semantic network of the sentence "Donokurai kakarimasuka ?" (an ambiguous question, when translated into English, it can mean "How much does it take ?" or "How long does it take ?").

```
(n0 take (form what-question)
         (wh-kind how-degree)
         (target ?))
```

In this example, "?" means that the value of the slot cannot be understood. "?" is resolved by the context processing part. However, when the system cannot resolve it, the system activates the clarification dialog.

We use the semantic network representation without a first node-name in order to denote the meanings in some system databases. This representation will appear in the following sections. The node-name with "$" represents the variable that can match any node-name or semantic network. For example, "(go (to $dist))" means that someone goes somewhere.

3.4 Context Processing

The context processing part performs the following two processes: finding pronoun antecedents and resolving ellipses.

3.4.1 Ellipsis Resolution by Default Value

The domain of our system is sightseeing guidance for Mt. Fuji. Since the domain is very restricted, it is possible to resolve some ellipses by default values. For example, the user input "How can I go from Toyohashi ?" is interpolated to "How can I go to Mt. Fuji from Toyohashi ?" as long as the system shows explicitly the domain to the user. The detailed discussion appears in Ref.(2).

3.4.2 Ellipsis Resolution by Context

Our system stores the semantic networks of previously inputted dialog sentences in the context stack. Ellipses are resolved by reference between the semantic network currently being processed and the networks in the context stack. Consistency is checked by comparing all values in the slots. If no slot contains different values, two networks are consistent. The omitted slot value is obtained from a consistent network in the context stack.

3.4.3 Resolution of the Target Slot of a Question

If a target slot name of a question, that is, a value of the "target" slot, is omitted, the system cannot answer the question. For example, the question presented previously "Donokurai kakarimasuka ?" is ambiguous referring to cost or time. Our system interpolates it by using two kinds of knowledge. One is the verb slot dictionary, and the other is the interrogative pronoun dictionary.[6]

The verb slot dictionary is a set of lists of a verb and the verb's applicable slot names. For example, the entry for the verb "iku" (go) is "(iku agent from to instrument cost time)". The elements in the parenthesis other than "iku" are slot names that "iku" can take. The interrogative pronoun dictionary is a set of lists of interrogative pronouns with applicable slot names. Japanese "what" questions need an interrogative pronoun and the pronoun constrains the target slot name of the question. The list represents a set of candidates for the target slot name of the question. For example, a list for "donokurai" (in English, "how much", "how long" or concerning with other quantity) is "(donokurai height mass cost time)".

The system can obtain candidates for target slot names by using the dictionaries. The intersection of two sets of candidates further restricts them. If there is more than one candidate, the system activates the clarification dialog.

3.4.4 Anaphora Resolution

The system employs very simple heuristics for

anaphora resolution. The antecedent of a pronoun is decided by a search of the context stack. The consistent node in the semantic network of the most recent utterance is regarded as an antecedent candidate. The consistency is checked by comparing semantic features used in the parser. Since this simple method often makes mistakes, the system activates a verification dialog to confirm the selection.

3. 5 Generating the Next Utterance

3. 5. 1 Outline

The next utterance generation part is invoked after processing the context of the input sentence. Figure 4 shows the flow of this part. If the user utterance is a question, this part invokes a new dialog rule for answering the question and interprets it. For a declarative sentence, the part executes an active dialog rule on the rule stack. The active dialog rule is the top rule of the rule stack. The user input is predictively interpreted by an active dialog rule. The mechanism can be used for an information-seeking task where a user gives the system the information that needs to be retrieved from a database and the system gives the user the information that s/he wants. However, the rules and databases developed in the following sections are, of course, specific to simulating the Mt. Fuji travel bureau. Although there is some research that uses rule sets for deciding the next utterance, our method is different from these[7] regarding the invocation mechanism of new dialog rules and the set of instructions.

Since in our system the user query invokes the dialog rules, we can easily develop the rules for the complex question-answer dialog. For example, the dialog rules for answering the user question can include the "ask" instruction for asking what the system needs. Also our system has a mechanism for dialog clarification and verification. In the processing of the user utterance, the invocation of dialog rules for clarification and verification occurs prior to normal dialog rules. Previous dialog systems make questions for verification by a special mechanism for each part. Our system invokes dialog rules for verification that are interpreted in the same way as normal dialog rules. Since our system can use all the facilities of the system for verification and clarification, we can easily develop complex rules for them.

3. 5. 2 Dialog Rules and Database for Sentence Generation

A dialog rule is a set of instructions. Figure 5 shows the form and instructions of a dialog rule. Each instruction is interpreted in downward order. Interpretation is terminated at the generation of interrogative or long assertive sentences. The following are explanations for each instruction.

If: Most instruction in a dialog rule belong to this type. An "if" instruction consists of a condition and instructions. If the condition is true, instructions are interpreted. Elements in the instruction part of "if" are instructions for the dialog rule. Of course we can use "if" instructions recursively. For the conditional part, two predicates are prepared. The "bel" predicate takes a semantic network as an argument and returns true when there is a consistent network to the argument in the context stack. The "input-now" predicate is the same as "bel" but matches the previous user input only. Also, logical operators such as "and", "or" and "not"

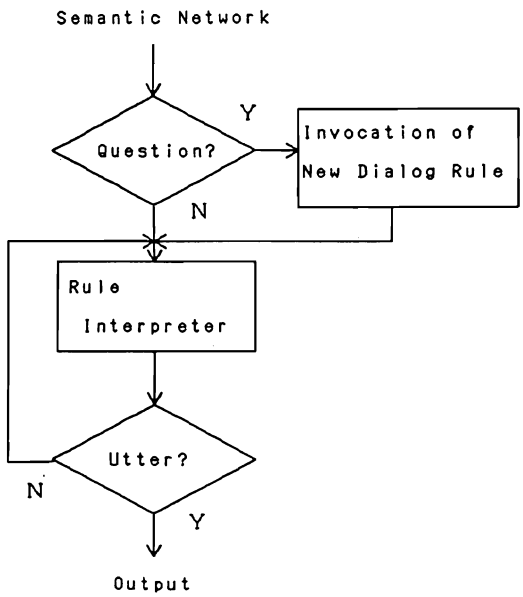

Fig. 4 Flow for generating the next utterance.

```
Syntax of dialog rule:
  (defrule rule-name
     instruction1
     instruction2
        ⋮              )

Instruction:
  (if condition instruction1 instruction2 ...)
  (tell SEM-net)
  (tell2 string)
  (ask type SEM-net)
  (call rule-name)
  (either (condition1 instruction1)
          (condition2 instruction2)
                  ⋮
                                    )
  (return)
  (s-return)

Condition:
  (bel SEM-net)
  (input-now SEM-net)
```

Fig. 5 Syntax and instructions of dialog rule.

```
(go  way
  const:  (from Toyohashi)(to Mt.Fuji)(by car)
  (go (to Gotenba)(way $x)):
    "You can use Route 138 from Gotenba."
  nil: "Go to Gotenba interchange by Toumei highway.
        From there you can use Route 138." )
```

Fig. 6 An example of Tell database.

can be used in the conditional part.

Tell: A "tell" instruction takes a semantic network as an argument and generates a system utterance. The semantic network must include a variable which refers to the content of the utterance. The sentences themselves are stored in the special database for "tell" instructions (tell database). The tell database includes almost all information that the user wants. The interpolated argument by context is used as a search pattern for the tell database. Slots without variables are constraints for the search.

Since sentences for explanation depend on the user knowledge level, sentences in the tell database have other conditions based on this knowledge level. The user knowledge level is known as the user model.

An example of the tell database is shown in Fig. 6. The first line shows that this entry is data for the route of a trip. The second line represents the constraints for the usage. The following lines are sentences for the explanation of the route of the trip. The semantic network and nil are user knowledge constraints. If the user knows a value of the variable in the network, the next sentence can be used for the explanation. "Nil" represents no constraint.

In a more general dialog system, this database corresponds to the database of answers in the system's domain. To facilitate modularity, the databases of answers and dialog control instructions should be separated.[8] However, this issue is not important for our purposes.

Tell2: A "tell2" instruction takes a string as the argument and generates it directly as the next system utterance.

Ask: An "ask" instruction is used for generating questions by the system. It takes two arguments. One is the same as the argument of "tell". The other represents the type of question such as "yes/no", "what", or "which". There is a database for "ask" instructions similar to the tell database. An output sentence is a question seeking a value of a variable in the semantic network. Moreover, the semantic network is stored in the predicted template for context processing.

Call: Invocation of a new dialog rule.

Either: A set of pairs of conditions and instructions. If the condition is true, pairing instructions are interpreted. All the conditions are evaluated at the same time. When many conditions are true or all the conditions are false, the system cannot decide on an instruction. In this case, the system activates the clarification dialog.

```
(defrule  how-to-go
  (if (not (bel (go (from $sour))))
      (ask what (go (from $sour))))
  (if (not (bel (go (by $inst))))
      (ask what (go (by $inst))))
  (tell (go (way $x))))
```

Fig. 7 An example of dialog rule.

```
(go (way (how-to-go))
    (instrument (rule1)))
```

Fig. 8 An entry of rule database.

Return, s-return: A dialog rule is terminated and the previous rule is popped from the rule stack. The popped rule is executed continuously. "Return" pops the rule stack only. But "s-return" also pops the system stack. "S-return" is used for the verification and clarification dialog rule.

An example of a dialog rule is shown in Fig. 7. This rule is used for teaching how to go. If the system does not know the departure point or the means of transport, the interpreter executes an "ask" instruction and generates a question. After the user response to any questions, the system generates an answer by the "tell" instruction.

A dialog rule is invoked by user questioning. The rule database stores the rule name accessed by the semantic network of the user question. The semantic network of a "what" question has a "?" as a slot value. An invoked rule is searched in the rule database by the class name of the network and the slot name with "?". Figure 8 shows an entry in the rule database. The rule of "how-to-go" is invoked by the network

(n_0 go (form what-question)
(wh-kind how-method)
(target way))

of the sentence "douyatte ikunodesuka ?" ("How do I go ?") and the rule database.

3. 6 Generation

The speech generation part generates a voiced utterance. Since our research concentrates on speech understanding and dialog, we implement a simple mechanism using templates. We have prepared some PCM recording templates for the system's representative sentences. The system interchanges some words in a template for generating different sentences from the representative sentences.

3. 7 Invocation of the Verification and Clarification Dialog

When ambiguities or uncertainties that cannot be resolved within the system occur, the system activates the verification or clarification dialog. The part where the ambiguities exist dictates the dialog rule and calls the part invoking the verification and clarification dialog. That part pushes the current processing status into the system stack and invokes the whole dialog system using the decided rules. Returning from the clarification dialog to the previous one is achieved by an "s-return" instruction. The restarted part can use the information received from the verification or clarification dialog.

4. Verification and Clarification Dialog for Each Part of the Dialog System

4. 1 Ambiguities in the Dialog System

Even a human asks questions to verify information when s/he cannot resolve ambiguities in another person's utterance. Since machines have a lower ability for resolving ambiguity than humans, it is important to aggressively ask verification questions.

A human processes an utterance by a unified mechanism consisting of, for example, speech recognition, morphology analysis, and parsing. Although there are studies to develop a unified mechanism for machine processing,[9] we employed separated mechanisms to process efficiently spoken sentences and dialog using current techniques. The parts of separated mechanism systems are isolated and connected by very narrow pipes. A human can resolve many ambiguities in spoken natural language based on whole kinds of knowledge, which, in comparison, a machine cannot do. The part of a machine dialog system must resolve an ambiguity using knowledge contained within that part and the results from a few other parts. If the part needs more information from other parts, a dead-lock problem where all parts must wait for the results of other parts may occur. However, the system can avoid this situation by using a clarification or verification query.

There are many methods for the clarification and verification in compliance with the processing in each part. In this section they are discussed.

4. 2 Speech Recognition

It is well known that speech recognition is a very difficult task. The reason is that a speech signal consists of many indeterminacies such as varieties of speech pattern depending on speaker differences, concatenation, or duration of utterance, ambiguities of segmentation, interjections, and noise. To cope with these indeterminacies, stochastic methods have been employed in a recent automatic speech recognition system.[10] The result of recognition by stochastic methods is a set of candidate strings with respective probabilities. If the maximum probability among the candidate strings is lower than a threshold, the system cannot determine a correct answer.

In the clarification dialog for the speech recognition part, it is important to decide what to ask and how to ask it. Although some studies have been conducted on this subject,[11] there has not yet been any investigation into a dialog system based on modern speech recognition technology.

The simplest method of clarification is to prompt the user to repeat the same sentence by outputting "wakarimasen" (in English "don't understand"). However, this may be the least efficient method of clarification. As stated in the Sect. 2, humans seldom ask to repeat complete sentences. The system should be able to ask to clarify only the ambiguous part of a sentence. Therefore only if the system obtains no candidates from the input sentence or candidates with very low probabilities, does the system say "I don't understand". Otherwise, our system shows some possible candidates to the user and requests the user to select the appropriate candidate.

4. 3 Sentence Comprehension

The sentence comprehension part consists of three processes: morphological analysis, parsing, and transformation into a semantic network. Although ambiguities can occur in each subpart, the system should not ask a question as soon as it occurs, because many ambiguities can be resolved by the cooperation of three parts. For example, the segmentational ambiguity of the morphological analysis subpart can be resolved by parsing with semantic features. It is also difficult to ask the user to clarify segmentational ambiguity using natural language.

The system leaves ambiguities of the three subparts alone until obtaining a semantic network from the sentence comprehension part. The result is a set of semantic networks as candidates. If the set has more than one candidate, the system activates a clarification or verification dialog. In our system the question for

```
(defrule verification-network
  (ask yes/no $sem-net)
  (if (or (input-now (exc_yes))
          (input-now (exc_no)))
      (s-return))
  (tell2 "hai mataha iie de okotae kudasai."))
         ("Please answer yes or no.")
```

Fig. 9 A dialog rule for verification.

verification is generated from a semantic network by a special template for semantic network patterns.

Figure 9 shows the dialog rule for the verification of a candidate with a low probability. "$sem-net" is replaced by the candidate semantic network. (Exc_yes) and (exc_no) are semantic networks of "yes" and "no", respectively. This rule is active until the user utters "yes" or "no".

4.4 Context Processing

Elliptical phrases are interpolated by the context stack and temporal template. In many cases, dialog rules use "ask" instructions to obtain information for answering the user's original question. The information from the user is stored in the context stack. This is used to resolve ellipses when our system answers the original user question. However, since our interpolating method is very simple, the system sometimes makes a mistake. Our system verifies interpolated sentences before answering the user's original question.

Since an anaphora resolution result is also uncertain in our system, the system must verify whether it is correct or not. Recognizing possible mistakes, the system prompts the user to utter a sentence without using pronouns.

4.5 Rule Invocation

The "either" instruction for dialog rules is used for the system to choose the appropriate dialog rule from alternatives. Each candidate dialog rule has a condition. The rule with a true condition is activated. If two or more rules are activated or all conditions are false, the system cannot decide the next activating rule. An ambiguity of rule selection occurs.

To generate a question for the selection, each rule has a sentence that expresses the rule's purpose. The system shows the user all the rule explanations and the user selects the appropriate rule.

4.6 User Model

We described the user knowledge level in Sect. 3.5.2. Explanation sentences that have the same meanings should be varied according to the user knowledge level. In response to the same input question, the system must respond with an answer more easily understood by a user with lower level knowledge about the domain than by an upper level user.[12]

The tell database has a sentence with a condition concerning the user's knowledge level. If there are alternative answers, the system can confirm the user's knowledge level directly.

```
sys: Fuji-san kankou annai-jo desu.
     (This is the Mt.Fuji travel bureau.)
usr: Fuji-san-e ikitai.
     (I'd like to go to Mt.Fuji.)
sys: Fuji-san-no kankouchi-ni tsuite setsumei
     shimasuka?[4.5]
     soretomo, Fuji-san-eno ikikata-wo
     setsumei shimasuka?
     (Shall I explain sightseeing spots near
      Mt.Fuji or the way to Mt.Fuji?)
usr: douyatte Fuji-san-made ikemasuka?
     (How can I go to Mt.Fuji?)
sys: dokokara kimasuka? [3.5.2]
     (From which city?)
usr: Toyohashi-kara ikimasu.
     (I'll go from Toyohashi.)
sys: kuruma-de kimasuka? densha-de kimasuka?
     [3.5.2]
     (Are you coming by car or train?)
usr: sokomade kuruma-de ikimasu.
     (I'll go there by car.)
sys: "soko" towa Fuji-san-no kotodesuka? [4.4]
     (Does "there" refer to Mt. Fuji?)
usr: hai.
     (Yes.)
sys: Toyohashi-kara Fuji-san-made kuruma-de
     iku houhou-wo otazune desuka? [4.4]
     (Are you asking how to go to Mt. Fuji from
      Toyohashi by car?)
usr: hai.
     (Yes.)
sys: Gotenba-madeno ikikata-wo shitte
     oraremasuka? [4.6]
     (Do you know how to get to Gotenba City?)
usr: iie.
     (No.)
sys: Toumei kousoku-de Gotenba intā-made kite,
     sokokara 138 gousen-de ikemasu. [3.5.2]
     (Go to Gotenba interchange by Toumei
      expressway, use Route 138 from there.)
usr: dorekurai kakarimasuka?
     (How much does it take?) or
     (How long does it take?)
sys: jikan-ga desuka? okane-ga desuka? [3.4.3]
     (Do you mean time or money?)
```

Fig. 10 Example of dialog.

5. An Example of Dialog

Figure 10 shows an example of a dialog between a human user and our dialog system. "Usr" and "sys" indicate the user and system utterances, respectively. The actual dialog is carried out in spoken Japanese. Sentences in the parentheses are the English translations. The number in brackets following system utterances shows the section in this paper where the generation mechanism is described.

6. The Current State of the System

In Sect. 2, we categorized clarification and verification queries. The categories of queries that the

current system can generate are (1), (3), (5) and (6) in Sect. 2. 2. The knowledge for generating the queries of (1), information seeking to answer questions, is stored in dialog rules as sequences of instructions (see Sect. 3. 5). The queries of (3), deciding the level of answer, are generated by executing a "tell" instruction with reference to dialog rules (see Sects. 3. 5. 2 and 4. 6). If the current system cannot obtain a result from the speech recognition part, parse a sentence or transform it into semantic networks, it generates a sentence indicating that it cannot understand the user input. This corresponds to category (5), representation of incomprehension. The other processes in Sect. 4 generate queries for verification of sentence meanings corresponding to category (6).

The queries of category (7) repeat part of or the entire previous sentence. From the viewpoint of verification of sentence meaning, category (7) is included in category (6). However, category (7) is concerned with the surface sounds of sentences. The system cannot simulate dialogs exactly like category (7), because it has to repeat the same surface sound, not the same meaning. The system does not take account of that, although the same surface sound is generated sometimes by chance. The mechanism for generating queries of (2), refinement of meaning, requires a method for deciding what the system asks from the input sentence. We need more studies of the method of deciding the direction of refinement. We will continue study in this field using our system. The queries of (4) and (8), clarification of word meaning and verification of sentence part sounds, cannot be generated using a simple speech recognition mechanism unable to cope with unknown words.

Since we did not experimentally evaluate the effectiveness of our clarification and verification mechanism, we cannot clearly judge its effectiveness. However, we can state with certainty that we can easily develop a system with clarification and verification for all parts, using our framework.

There are many studies regarding the refinement of sentence comprehension ability. If powerful techniques such as the plan-based mechanism for inference of user intention are used and the system makes a correct inference, it is a great advantage for the user. However, if the system misunderstands a sentence, the subsequent dialog may collapse or the system may generate an answer not intended by the user. If the confidence in results can be measured, we can develop a better system that employs powerful techniques and clarification and verification methods. Unfortunately, it is difficult to measure the confidence in results when using such powerful techniques.

Although we do not intend for the system to simulate a human dialog and we have developed the system that can make man-machine dialog more reliable, it is, of course, important that it be user-friendly. Too many verification and clarification queries tend to irritate the user. The reader might think that the disadvantages of too many queries outweigh the advantages of a natural language interface. However, the user's freedom for input and the lack of prior learning requirements, which are important advantages of a natural language interface, are retained as features of our system with clarification and verification.

Our dialog system forms many questions that require only a "yes" or "no" response, which is easily recognizable by the current speech recognition mechanism. Thus it is important to study ways of querying which limit the user's response.

7. Conclusions

We have described a spoken dialog system that can generate verification and clarification queries. In human-human dialog, ten percent of all sentences collected are concerned with verification or clarification. In human-machine dialog, it is even more important that the machine verifies ambiguities that occur in processing a dialog. In particular, since speech signals have many uncertainties, the clarification dialog is very important.

The verification and clarification functions of our spoken dialog system are implemented using all facilities of the dialog system recursively. We have confirmed that our system is more robust than systems having no verification or poor verification and clarification functions.

Our current research direction includes extending verification and clarification methods to facilitate more natural man-machine communication.

Acknowledgements

The authors wish to thank Mr. Masatoyo Taguchi, who developed an earlier version of the dialog system, and Mr. Atsuhiko Kai, who developed the basic speech recognition system.

We used transcripts in the continuous speech corpus for research of the Acoustical Society of Japan. We wish to thank all members of the Committee on Continuous Speech Database of the Acoustical Society of Japan.

References

(1) ed. Brady, M. and Berwick R. C., *Computational Models of Discourse,* MIT Press, 1983.

(2) Nakagawa, S., Takemoto, S. and Taguchi, M., "Translation from Japanese sentence to first order predicate calculus in question-answering system for traffic regulation LICENCE," *Trans. IPSJ,* vol. 32, no. 3, pp. 354-363, 1991.

(3) Niedermair, G. T., Streit, M. and Tropf, H., "Linguistic processing related to speech understanding in SPICOS

II," in *Speech Comm.*, 9, pp. 565-585, 1990.
(4) Kai, A. and Nakagawa, S., "Consideration on HMM-based continuous speech recognition using word spotting methods," *IEICE Technical Report (Speech)*, SP92-10, 1992.
(5) Yamamoto, M., Kobayashi, S. and Nakagawa, S., "An analysis and parsing method for the omission of post-position and inversion on Japanese spoken sentences in dialog," in *Proc. of the symposium on new applications of natural language processing*, pp. 86-93, 1992.
(6) Yamamoto, M., Kobayashi, S. and Nakagawa, S., "A query generation system to disambiguate meaning or sentences in dialog," *Proceedings of the 5th Annual Conference of JSAI*, 13-8, pp. 551-554, 1991.
(7) Yamamoto, H., Kai, K., Osato, M. and Shiino, T., "Design of an intelligent CAI system for training a foreign language based on simulation of conversation," *Trans. IPSJ*, vol. 30, no. 7, pp. 908-917, 1989.
(8) Yoshida, H., Yamamoto, T., Nomura, Y., Yamashita, Y. and Mizoguchi, R., "Utilization of the dialog flow in dialog management system MASCOTS," *Proceedings of the 5th Annual Conference of JSAI*, 13-2, pp. 527-530, 1991.
(9) Hasida, K. and Takezawa, T., "Aspects of integration in natural language processing," *Computer Software*, vol. 8, no. 6, pp. 3-16, 1991.
(10) Nakagawa, S., "Speech Recognition Based on Stochastic Model," *IEICE*, 1988.
(11) Ukita, T., Ishikawa, N., Nakagawa, S. and Sakai, T., "Confirmation methods of utterances in a dialog system by speech," *Trans. IPSJ*, vol. 22, no. 6, pp. 589-595, 1981.
(12) Wahlster, W. and Kobsa, A., "User models in dialog systems," *User Models in Dialog Systems*, A. Kobsa and W. Wahlster (ed.), Springer-Verlag, pp. 4-34, 1989.

Mikio Yamamoto was born in Kumamoto, Japan, in 1961. He received the B.E. and M.E. degrees in information and computer sciences from Toyohashi University of Technology, in 1984 and 1986. He joined the OKI Techno Systems Laboratory, Inc, Nagoya, Japan from 1986 to 1988. He is a technical official in Toyohashi University of Technology. Mr. Yamomoto is a member of the Information Processing Society of Japan, Japanese Society of Artificial Intelligence and American Association for Artificial Intelligence.

Satoshi Kobayashi was born in Yamanashi, Japan, in 1966. He received the B.E. degree in information and computer sciences from Toyohashi University of Technology, Aichi, in 1991. Since 1992, he has been a student of Master course of the Information and Computer Sciences, Toyohashi University of Technology. He is engaged in research in Natural Language Processing.

Yuji Moriya was born in Aichi, on February 22, 1969. He received the B.E. degree from Toyohashi University of Technology, Aichi, in 1991. Since 1991, he has been a student of Master course of Information and Computer Sciences, Toyohashi University of Technology. He has been engaged in research on the speech recognition and understanding systems. Mr. Moriya is a member of Information Processing Society of Japan.

Seiichi Nakagawa was born in Kyoto, Japan, in 1948. He received the B. E. and M.E. degrees in electrical engineering from Kyoto Institute of Technology, Kyoto, in 1971 and 1973, respectively, and Dr. Eng. degree from Kyoto University, Kyoto, in 1977. He joined the Faculty of Kyoto University, in 1976, as a Research Associate in the Department of Information Sciences. From 1980 to 1983 he was an Assistant Professor, from 1983 to 1990 he was an Associate Professor and since 1990 he has been a Professor in the Department of Information and Computer Sciences, Toyohashi University of Technology, Toyohashi. During 1985 to 1986, he was a visiting scientist in the Department of Computer Science, Carnegie-Mellon University, Pittsburgh, USA. He is an Author of the book "Speech Recognition Based on Stochastic Model" (in Japanese). Inst. Elect. Inform. Comm. Engrs. Japan, 1988. Dr. Nakagawa was a coreceipient of the 1977 Paper Award from the IEICE and the 1988 JC Bose Memorial Award from the Institution of Electro. Telecomm. Engrs.