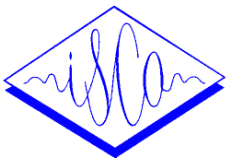

2nd International Conference on
Spoken Language Processing (ICSLP 92)
Banff, Alberta, Canada
October 12-16, 1992

AN INFORMATION RETRIEVAL SYSTEM WITH A SPEECH INTERFACE

Yasuhisa Niimi and Yutaka Kobayashi

Kyoto Institute of Technology
Matsgasaki, Sakyo-ku, Kyoto, 606 Japan

ABSTRACT

This paper describes a speech interface to an information retrieval system. It consists of three main components; speech recognition system, command generator and response generator. The speech recognition system accepts a spoken command of a Japanese sentence and passes the recognized sentence to the command generator, which translates it into a formal query command to operate the information retrieval system. The response generator receives retrieved data and produces a response to the user in a written sentence. We proposed a basic strategy in construction of the speech recognition system. It is that the top-down linguistic hypotheses are made at the lexical level while they are verified by using units independent of the word, phonetic strings bounded by robust phones (phones which can reliably be detected) in order to reduce the misrecognition of short function words. In the natural language interface, syntactic and semantic analyses are simultaneously performed. This makes it possible to resolve syntactic ambiguities. The interface was tested by using the speech corpus of 53 sentences spoken by each of three male speakers. The most promising rate of sentence understanding was 89.9 % for a small task.

10.21437/ICSLP.1992-376

1. INTRODUCTION

Speech recognition based on the hidden Markov model has made a great deal of progress and advanced LSI technology has made it possible to implement speech recognition systems capable operating in real time. For applications of this technology to several specific task domain, man-machine dialogue systems have been developed[1-7].

In this paper we describe a speech interface to a database. The database includes information of sightseeing in some city. The speech interface consists of a speech recognition system and a natural language interface to an information retrieval system. The natural language interface is composed of a command generator and a response generator. The speech recognition system accepts a spoken command of a Japanese sentence and passes the recognized sentence to the command generator, which translates it into a formal query command to operate the information retrieval system. Using retrieved data, the response generator produces a response to the user in a written sentence.

Japanese has several short words like postpositions and auxiliary verbs. The misrecognition of these types of words cannot be restored by the latter semantic processing. In order to avoid this kind of errors, we proposed a basic strategy that top-down linguistic hypotheses are made at the lexical level while they are verified by using units independent of the word, phonetic strings bounded by robust phones. The robust phone is a phone, such as an unvoiced fricative or a short silence before unvoiced plosives, which can reliably be detected. This makes it possible to cope with inter-word and intra-word coarticulation in the same way.

In the natural language interface, syntactic and semantic analyses of recognized word strings are made simultaneously. There is a semantic rule corresponding to each of syntactic rules described in the definite clause grammar. Syntactic rules are used to parse substrings of words into phrases and semantic ones to make semantic interpretations of parsed phrases. Simultaneous applications of both kinds of rules resolve some syntactic ambiguities included in an input word string. When the input word string can be parsed as a sentence, a semantic interpretation of the sentence, that is, a formal query to the database, has been completed.

In section 2 we describe the task domain to which the speech interface is applied, including a task-specific grammar. In section 3 we give a brief explanations of the speech recognition system. Then we explain the language processor with an emphasis on the semantic processing in section 4, and finally report some experimental results.

2. TASK SPECIFICATION

2.1 Database

We suppose that the user of the speech interface could issue oral questions about the contents of a relational database, and get responses as written sentences. The database contains information on temples, shrines, museums, universities and hotels in Kyoto city. The relational database consists of tables like the one shown in Fig. 1. The name of each table is considered as a relation. A column contains values of relation components, called attributes. A row corresponds to a record, an instance of the relation, which is a list of the values of the attributes.

attribute name 1	attribute name 2	...
attribute value 11	attribute value 21	...
attribute value 12	attribute value 22	...
...	...	...

Fig. 1 A relational Table.

(a) Kosendai no denwabangou wo oshietekudasai.
(Please tell me the phone number of Kosendai ?)
(b) Kinkakuji no haikanryo wa hyakuen desuka.
(Is the entrance fee of Kinkakuji temple 100 yen ?)

Fig. 2 Example of acceptable sentences.

2.2 Task syntax and semantics

Fig. 2 shows a few Japanese sentences of sightseeing queries. Fig. 3 illustrates syntactic rules for query sentences by the definite clause grammar (DCG). The terminal symbols of this grammar are underlined and the nonterminal ones are not. The symbols in the

parentheses are semantic markers. The terms enclosed by the brackets expresses the predicates for the semantic constraint. The vocabulary contains 248 words. The test set perplexity is 8.3 for the current task.

```
s ---> cp(C),fn(wa),ppc(C,q) ¦ cp(C),fn(wo),v(find)
       ¦ tp(T),fn(wa),ppt(T,q) ¦ tp(T),fn(wo),v(find)
tp(T) ---> test(T) ¦ {cat(T,C)},cp(C),fn(hap),tn(C,T)
       ¦ {cat(T,C)},cp(C),{verbf(C,F,T,V)},fn(F),v(V,n)
cp(C) ---> cst(C) ¦ cn(C),uch,ppc(C,rt),mono
       ¦ ppc(C),cn(C)
ppc(C,q) ---> {cat(T,C)},qn(T),{verbf(T,F,C,V)},fn(F),
              v(V,q)
ppt(T,S) ---> cfp(T,F),adj(T,F,S) ¦ dp(T),aux(T,S)
sp(T,C) ---> tn(T,C),fn(subj)
cfp(T,F) ---> tp(T),{compf(T,F)},fn(F)
dp(T) ---> tp(T) ¦ cfp(T,F),an(T,F)
```

Fig. 3 A part of syntactic rules of the sightseeing task.

2.2.1 Syntactic structures and categorization of words

The nonterminal symbols describing the task grammar reflect the specific features of the database queries. Besides the starting symbols s, we introduced seven nonterminals.

- cp: noun phrases which specify the search key in the relational tables.
- tp: noun phrases which specify the attributes in the relational tables.
- ppc, ppt: predicative phrases containing a verb which terminates a sentence or modifies a noun phrase. ppc modifies cp and ppt does tp.
- sp: noun phrases which specify the main topic in a ppc phrase.
- cfp: phrases which represent an object of comparison.
- dp: noun phrases which specify case fillers in ppc or ppt phrases.

We categorized the words in the vocabulary into fourteen task specific categories. Especially the nouns were subcategorized according to the concepts related to the database. The words other than the noun were classified according to the standard school grammar. The categorization of the noun is described below.

- cn: names of relational tables in the database.
- tn: names of attributes in a relational table in the database.
- cst: proper nouns which appear in the column of the name attribute.
- tcst: nouns which appear in the columns other than the name attribute.
- qn: interrogative nouns.

2.2.2 Semantic markers and case structure

In order to judge whether or not a particular word pair may appear in a particular syntactic structure, we introduced parameters playing roles of semantic markers to the syntactic rules. Semantic constraints between noun phrases can be classified into two categories in the present task.

(1) "A no B (B of A)" imposes a constraint that B is an attribute name of a record specified by the noun phrase A. Both specified items must exist in the same relational table.

(2) "C wa D desuka (Is C D ?)" implies that C and D refer to an attribute name and its value, respectively. Both C and D must be relevant to the same attribute of the database, for example, "time" and "8:30".

We use semantic markers C and T for describing these constraints. The syntactic categories cn and cst are given the marker C as a parameter whose value is one of the names of relational tables. The syntactic categories tn and qn are given the marker T as a parameter whose value is one of higher concepts of the attribute names. The predicate {cat(T,C)} in Fig. 3 expresses the constraint that the value of T be an attribute of the relational table specified by the marker C.

We describe the semantic (or co-occurrence) relation between an adjective or a verb and other phrases by the semantic marker and the case grammar. In the present task the case frame of both an adjective and a verb has two case slots. Syntactically one of them acts as the subject of a sentence. Since the predicative phrase including an adjective is usually interpreted as a comparison operator, two case slots must filled by phrases with the same semantic marker. Thus the adjective can be characterized by a semantic marker. The slot filler other than the subject of a sentence forms the predicative phrase with a postposition and an adjective itself. This co-occurrence relation between the adjective and the postposition can be tabulated. The predicate {compf(T,F)} in Fig. 3 expresses such a relation, where the parameter F denotes the class of postpositions.

In the case of verbs the two slot fillers are described by different semantic markers; one by T and the other by C. The predicate {verbf(T,F,C,V)} in Fig. 5 expresses a relation among these two semantic markers T and C, the class of verbs (denoted by V), and the postposition which can be attached to the slot filler other than the subject.

3. THE SPEECH RECOGNITION SYSTEM

The speech recognition system is composed of five components: acoustic processor, matcher, phonological and linguistic processors, and controller. The acoustic processor performs signal processing based on the linear predictive coding (LPC). For each 10 ms of the speech the processor calculates the LPC-cepstral coefficients (LPC-CEPs), their time derivatives (DELTA-CEPs), and a pair of short-term energy and its time derivatives (POWs), and then vector-quantizes them separately.

The speech recognition system has three different lexical matchers, called lattice-matcher, HMM-matcher and STA-matcher[8,9]. The acoustic processor provides each of these matchers with different outputs. For example, it provides phonetic lattices for the lattice-matcher. The processor identifies 17 broad phonetic classes by using the discrete phoneme-based hidden Markov models with three states, and produces a phonetic lattice, which is called an input lattice below. The broad phonetic classes which are currently identified are five vowels, two semivowels, nine consonant classes and a silence. Of these broad phonetic classes, the unvoiced fricative and the silence are used as the robust phone since they can reliably be detected.

The linguistic processor makes top-down hypotheses, each consisting of a set of words capable of following a partial sentence selected by the controller. A partial sentence is a string of already recognized words covering the beginning portion of an input utterance.

The phonological processor computes phonetic variants and the standard duration of a word. Given the phonetic representation of a partial sentence and one of the words which can follow it, this processor applies phonological rules to the word with the last portion of the partial sentence as a phonetic context, resulting in a lattice describing phonetic variants of the given word. We call this lattice a phonetic lattice below.

We explain the lattice-matcher as an example of the matchers. Given a phonetic lattice bounded by two robust phones in the predicted word and the current segment of the input lattice, the lattice-matcher first looks for candidate robust phones in the input lattice within twice the standard duration of the phonetic lattice. After deciding a few candidate intervals, the matcher calculates the distance between both lattices using the lattice vs. lattice DP matching algorithm[10]. The distance is returned to the controller with the end

segment of the matched interval.

The controller invokes other components while extending likely partial sentence hypotheses. A partial sentence is a string of words which have been identified in the beginning portion of an input utterance. The controller preserves a set of partial sentences in the form of a tree. The neighborhood of leaves of the search tree is not usually verified. The controller selects one of these parts and makes the linguistic processor compute the words which can follow it.

4. NATURAL LANGUAGE INTERFACE

4.1 The outline of the natural language interface

The speech recognition system gives to the command generator a string of words as its output. The command generator analyzes the string syntactically and semantically, and transforms it into an internal representation. Although the speech recognition system makes the syntactic and semantic analyses of an utterance, their purposes are to make top-down hypotheses in order to reduce the search space in recognizing utterances.

Queries described in a natural language are generally classified into the two forms as shown in Fig. 4. The sentence (a) in Fig. 4 requires a value of an item of a record in some relational table, while the sentence (b) requires 'yes' or 'no' as an answer. Given these answers from the database system, the response generator produces a response to the speaker.

(a) Kinkakuji no haikanryo wa ikura desuka ?
(How much is the entrance fee of Kinkakuji temple ?)

(b) Kinkakuji no haikanryo wa 500 en desuka ?
Is the entrance fee of Kinkakuji temple 500 yen ?)

Fig. 4 Two types of query sentences.

```
sentence --> {function, quadruplet}
             |{function, quadruplet, quadruplet}
function --> find | operator
operator --> = | == | < | > | =< | >= | /==
quadruplet --> [C, N, T, V]
C   --> SM1
N   --> SM1 | quadruplet
T   --> SM2
V   --> SM2 | quadruplet | expression
expression --> (operator variable1 variable2)
SM1 ---> value of the semantic marker C | null
SM2 ---> value of the semantic marker T | null
```

Fig. 5 Syntax for the semantic representation of a sentence.

4.2 The internal representation of an utterance

The syntax for the internal representation is shown in Fig. 5. An internal form of an utterance is represented by a function and one or two quadruplets, that is, lists of four terms. The function indicates an action of the information retrieval system, corresponding to either of the two forms of the query sentences shown in Fig. 4. The functions corresponding to these two are 'find' for the query (a) and '==' (equal to) for the query (b). Other eight operators, '/==' (not equal to), '>', '<', '>=', '<=', and '=' (substitution) are used for the latter.

The quadruplet is a list of the following four terms.

(a) C: a name of a relational table of the database.
(b) N: an indicator of a record of the relational table.
(c) T: an attribute of the record.
(d) V: a value of the attribute.

Three of these four terms are extracted from an input query to construct a query command to retrieve the rest one. The terms N and T can be defined by a quadruplet as shown in Fig. 5. This means that a query sentence can include embedded sentences.

4.3 Semantic analysis

The semantic definition of a word is given by an incomplete quadruplet, a function, or a combination of both. An incomplete quadruplet is the one in which some of four terms are not specified. Examples of the semantic definitions of words and terminal symbols are shown in Fig. 6.

```
cn  --> [C,_,_,_]     ("shaji"  --> [temple,_,_,_])
tn  --> [_,_,T,_]     ("haikanryo" --> [_,_,fee,_])
cst --> [C,N,_,_]     ("Kinkakuji" -->
                              [temple,Kinkakuji,_,_])
adj --> {operator,T} ("yasui"  --> { >,fee})
```

Fig. 6 Semantic definitions of terminal symbols and words.

Each nonterminal of the grammar has a semantic form similar to the semantic definition of a word. There is a semantic rule corresponding to a syntactic rule. It gives a way to construct the semantic form of the nonterminal at the left hand side of the syntactic rule from those of the syntactic categories at the right hand side. Both a semantic rule and a syntactic rule are applied simultaneously during the linguistic processing. Some examples of the semantic rules are shown in Fig. 7.

```
(1) s({find,[C,N,T,V]}) --> tp([C,N,T,V]),fn,v({find})
(2) tp([C,N,T,V]) --> cp([C,N,_,_]),fn,tn([C,_,T,_]),
                      {nonvar(N)}
```

Fig. 7 Some examples of the semantic rules.

The first semantic rules states that the semantic form of a nonterminal s is {find,[C,N,T,V]} and is composed of an incomplete quadruplet [C,N,T,V], which is the semantic form of tp, and the operator {find} of v. The second rule declare following things: (1) the first and second parameters of the semantic form of tp is the first and second ones of cp respectively, (2) the third parameter of tp is the third one of tn, and (3) the first parameters of cp and tn should be equal to each other and the second parameter of cp should be definite (the predicate 'nonvar(N)' is true when N is filled with some value).

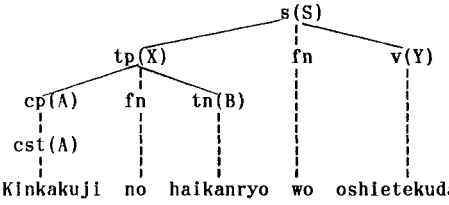

Fig. 8 The analysis of sentence "Kinkakuji no haikanryo wo oshietekudasai".
A = [temple,Kinkakuji,_,_]
B = [temple,_,fee,_]
X = [temple,Kinkakuji,fee,_]
Y = {find}
S = {find, [temple,Kinkakuji,fee,_]}

Fig. 8 shows the process of the syntactic and semantic analyses of a sentence "Kinkakuji no haikanryo wo oshietekudasai. (Please tell me the entrance fee of Kinkakuji.)" The two words "Kinkakuji" and "haikanryo (the entrance fee)" have the semantic forms [temple, Kinkakuji,_,_] and [temple,_,fee,_] respectively. According to the second rule shown in Fig. 7, these forms are combined to build the semantic form of the phrase "Kinkakuji no haikanryo (the entrance fee of

Kinkakuji)", resulting in the form [temple,Kinkakuji, fee,_]. Finally the first rule is applied to produce the semantic form of the sentence, which is {find, [temple,Kinkakuji,fee,_]}.

4.4 Generation of responses

The information retrieval system returns a complete quadruplet when the function of the query command is "find", and "yes" or "no" when it is a comparison operator. The semantic representation of a query sentence and this returned value are passed to the response generator. It has a set of patterns of the response, each corresponding to the pattern of query sentence. Fig. 9 shows examples of the patterns of the response. The response generator generates a response using these patterns and retrieved information.

Consider the query sentence, "Kinkakuji no haikanryo wo oshiete kudasai". Its semantic representation is {find, [temple,Kinkakuji,fee,_]} as shown in Fig. 8. The response pattern for it is "N no tn(T,C) wa V desu. (the tn(T,C) of N is V.)", the first pattern in Fig. 9. The execution of the query command {find, [temple,Kinkakuji,fee,_]} returns a complete quadruplet which gives V (for example, 500 yen) as the value of its fourth term, while N and tn(T,C) in this patter are known through the semantic analysis shown in Fig. 8. Thus the response for this sentence is "Kinkakuji no haikanryo wa 500 yen desu. (The entrance fee of Kinkakuji is 500 yen.)"

(1) {find, [C,N,T,V]} (V is unknown.)
response: "N no tn(T,C) wa V desu"
(2) {find, [C,N,T,V]} (N is unknown.)
response: "tn(T,C) ga V dearu C wa N desu"
(3) {op, [C1,N1,T1,V1], [C2,N2,T2,V2]}
response:
"N1 no tn(T1,C1) wa {N2 no tn(T2,C2)|V2}
yori adj(op,T) {desu|dewaarimasen}"

Fig. 9 Examples of patterns of the response.

5. EXPERIMENTS AND DISCUSSION

This section describes experiments carried out to test the speech interface explained in the previous sections. The text composed of 53 sentences was designed according to the syntax stated in the section 2. The speech corpus used in the experiments consists of these sentences read by each of three male speakers, namely, KB, NY and HA. The corpus contains about 9.4 minutes of speech in total.

For each speakers, eleven sentences from the first half was used in order to design three separate vector-quantization codebooks for the LPC-CEPs, the DELTA-CEPs and POWs. For the sentence recognition, the speech corpus of each speaker was divided into two halves. Except the codebooks, the parameters obtained from the initial half were used to recognize the latter half, and vice versa.

Table 1 The rates of sentence understanding

matcher	correct (%) first	-fifth
Lattice	83.0	93.1
HMM	88.7	92.2
STA	89.9	94.3

Table 1 shows the sentence understanding rates obtained for three different matchers. The columns, first and -fifth, stand for the rates of the correct sentences out of the 53 sentences in the first position and within the top five, respectively. The figures are the average rates for three speakers. The command generator not only translated all the correctly recognized sentences into intended query commands, but also correctly converted mis-recognized sentences which were semantically understood. The figures in Table 1 contains such sentences.

Several errors of the recognition occurred at very long compound words, such as /umekojijokikikanshakan (/umekoji/, /joki/, /kikansha/, /kan/), but few errors at short functional words. Moreover, most of the latter errors were relieved by the semantic interpretation in the command generator. The current phonological processor cannot decompose a long compound word into several component words to insert optional pauses between them, although this often happens in actual utterances.

6. CONCLUSION

This paper has reported the speech interface to a database and experiments carried out to test it. In construction of the speech recognition system, we proposed a basic strategy that top-down linguistic hypotheses are made at lexical level while they are verified by using units independent of the word, phonetic strings bounded by robust phones.

For 53 sentences read by each of three male speakers, the average rates of sentence understanding were 83.0 %, 88.7 % and 89.9 % for the three matchers, Although these figures are not sufficient for the practical use, most errors occurred at long compound words, but few errors at short functional words. We think mis-recognition of long words is more tractable than that of short words. This shows that the proposed strategy have been successful and is promising.

REFERENCES

[1] D.S.Pallet, J.G.Fiscus, and J.S.Garofolo, "DARPA Resource Management Benchmark Test Results June 1990," Proc. on Speech and Natural Language Workshop, pp.298- 305 Morgan Kaufmann Pub. (1990).

[2] N.Carbonel and J.M.Pierrel, "Task-oriented dialogue processing in human-computer voice communication" Recent Advances in Speech Understanding and Dialog Systems, pp.491-496 Springer-Verlag (1988).

[3] K.Matrouf et al.,"Adaptive probability-transition in DP matching process for an oral task-oriented dialogue," Proc. of ICASSP, pp.569-571 (1990).

[4] A.I.Rudinicky, "The Design of Voice-Driven Interfaces," Proc. of Speech and Natural Language Workshop, pp.120-124 Morgan Kaufmann Pub. (1989).

[5] S.J.Young and C.E.Proctor, "The design and implementation of dialogue control in voice operated database inquiry systems," Computer Speech and Language, Vol.13, No.4, pp.329-353 (1989).

[6] S.R.Young et al., "High Level Knowledge Sources in Usable Speech Recognition Systems," Comm. of ACM, Vol.32, No.2, pp.183-194 (1989).

[7] V.Zue et al., "The Voyager Speech Understanding System: Preliminary Development and Evaluation," Proc. of ICASSP, pp.73-76 (1990).

[8] Y.Kobayashi et al., "SUSKIT-II ---- A Speech Understanding System Based on Robust Phone Spotting," IEICE Trans. Vol.E-74, No.7, pp.1863-1869 (1991).

[9] Y.Kobayashi and Y.Niimi, "Segmented Trellis Algorithm for the Continuous Speech Recognition," in this volume.

[10] Y. Kobayashi and Y. Niimi,"Matching algorithm between a phonetic lattice and two types of templates --- lattice and graph," J. Acoust. Soc. Jpn., Vol.E5, No.4, pp.267-270 (1984).