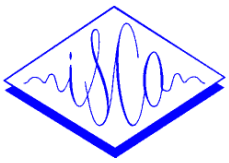

2nd International Conference on
Spoken Language Processing (ICSLP 92)
Banff, Alberta, Canada
October 12-16, 1992

A POWERFUL DISAMBIGUATING MECHANISM FOR SPEECH UNDERSTANDING SYSTEMS BASED ON ATMS

Shingo Nishioka, Yoichi Yamashita and Riichiro Mizoguchi

The Institute of Scientific and Industrial Research, Osaka University.
8-1 Mihogaoka, Ibaraki, Osaka 567 Japan.

ABSTRACT

A speech understanding system confronts with the ambiguities caused by the acoustic-phonetic errors and multiple-meaning of words. Thus the effective framework is required to resolve the ambiguity. In this paper we propose a generic framework for speech understanding system, which efficiently resolves the ambiguity of input.

I. INTRODUCTION

A speech understanding system confronts with the ambiguities caused by the acoustic-phonetic errors and multiple-meaning of words. In comparison with a natural language understanding, the input ambiguity is much larger, since natural language understanding assume only multiple-meaning of words. Thus the effective framework is required to resolve the ambiguity. In this paper we propose a generic framework for speech understanding system, which efficiently resolves the ambiguity of input. This framework is constructed based on a generic problem solving system with hypothetical reasoning.

The speech understanding system deals with local dependencies and global dependencies separately to avoid combinatorial explosion. Furthermore, the inference system provides the strong way to avoid the duplicated or useless inference under ATMS's [1] control. In these features, the speech understanding system can reduce the input ambiguity efficiently. Speech understanding in our system is formalized as a search problem under control of ATMS [2]. This framework is designed to reduce the overheads caused by ATMS, so the framework can be applied to other problems.

II. LIMITED GRAMMAR

In this chapter, we describe the limitations of the grammar we introduced.

As spoken language must be understood in real-time, its syntactical structure cannot to be too complex. For example, the deeper the sentence nests, the more effort is needed to understand it. Such a sentence is not appropriate for the spoken language. Thus syntactically complex sentences are hardly used in ordinary speech. So we decided to discard too complex sentences.

2.1 Complex Sentence and Compound Sentence

If the component of complex sentence is compound sentence, syntactic/semantic ambiguity occurs. To determine the structure is essentially difficult as it can be interpreted in two ways. For example, the sentence "SYOKUZIWO TUKURI, TEWO ARATTA KAASANWA, SOOZIWO SIMASITA" in Japanese can be interpreted in following two ways. (1) My mother cleaned the room, who made the lunch and washed her hands. (2) Someone made the lunch, and my mother cleaned the room, who washed her hands. Although the first interpretation is very popular, the second one can be a correct interpretation, too. In real life, such sentences are spoken with special intonation. Currently, our speech recognition system [3] does not recognize pauses, intonation and accents. So, we do not deal with such sentences.

2.2 Length of Noun Phrase

If a noun phrase which consists of only nouns, adjectives, RENTAISIs and adverbs becomes long in BUNSETUs†, the distance between the modifier and the modified noun becomes large. This increases the difficulty of understanding. We think these kind of complexity will not appear in spoken language, neither. Therefore, we do not deal with noun phrase more than 5 BUNSETUs. Of course, we can easily change the limit "5" by only changing the parameters of the speech understanding system. Thus, this limitation is not essential for the speech understanding system.

2.3 Summary of the Grammar

The limitations introduced are summarized as follows:
(1) a compound sentence does not appear as a component of a complex sentence.
(2) the size of noun phrase does not come up to 6 in BUNSETUs.

These limitations are not so strong for the spoken language. On the other hand, the limitation caused by the grammar for the written language is more serious for us. For example, the grammar cannot accept sentences which had been uttered incompletely, abbreviated, or do not have proper syntax.

III. SPEECH UNDERSTANDING SYSTEM

The input of speech understanding system is a BUNSETU candidate lattice which is generated from the output of speech recognition system according to the lexical grammar. Speech understanding system identifies a correct combination of BUNSETUs which satisfy the semantic constraints. Fig.1 shows an example of the input lattice for the speech under-

† A BUNSETU is the most basic and the shortest phrase for Japanese. It consists of a content word and function words. The function words indicate the role of the BUNSETU in the sentence.

10.21437/ICSLP.1992-408

standing system. We assume that the input sentence is spoken in each BUNSETU, so each column corresponds to the candidates for one BUNSETU. For example, the first column in Fig.1 (MAKURANOWO, MAKURANO ...) shows the candidates of the first BUNSETU in the speech.

Output of Speech Recognition System =
MAKURANO UENIWA TOUSANGA TIISANA BEDNO UEKAKENO KIRUTOWO KAKEMASITA
[My father put the kilt used as the bedcover on the pillow]

BUNSETU Candidates for the Input :

1st	2nd	3rd	...	8th
MAKURANOWO	UDENIWA	TOOSANGA	...	AKEMASITA
MAKURANO	UENIWA	TOOSANWA	...	TATEMASITA
MAKURAMO	GENNIWA	TOOTTA	...	KAKEMASITA
MATUDANO	GUMENIWA	TOOSANNOGA	...	SASEMASITA
MATUDANO	UDEKIWA	TOOSANKARA	...	MAKEMASITA
MAKURANOE	MENNIWA	OTOTTA	...	KAKEDASITA
...	...	...	...	...

Fig.1: Input for the Speech Understanding System

In this chapter, we describe the strategy to resolve the ambiguity of inputs efficiently.

3.1 Overview of the Speech Understanding System

The speech understanding system must cope with the difficulties caused by ambiguity in the utterances. Though there are many ways to enhance the system ability, in this paper, we decided to do it by employing powerful search strategies.

Speech understanding can be regarded as a searching problem in the space of the phoneme sequence with ambiguity. In general, there is control and inference in search. The inference subsystem verifies the validity of combination of BUNSETUs in the syntactic and semantic senses, because the most primitive unit is BUNSETU in our system as mentioned above. If the inference fails, the control subsystem determines the new combination to be verified.

If the inference subsystem is less affected by the control subsystem, it is easier to design appropriate searching method and controlling strategies. It is important to decide the appropriate size of data for control to reduce the controller's overheads. We designed the speech understanding system paying special attention to (1) reducing the overheads on controlling the system, and (2) using the efficient search mechanism.

Before considering about the above issues, we describe the concept of "clause" here. Let us consider about a cluster consisting of some BUNSETUs relating to each other. For example, a sentence "MAKURANO UENIWA TOOSANGA TIISANA BEDDONO UEKAKENO KIRUTOWO KAKEMASITA" [My father put the kilt used as the bedcover on the pillow.] has a structure like "((MAKURANO UENIWA) TOOSANGA (((TIISANA BEDDONO) UEKAKENO) KIRUTOWO) KAKEMASITA)". We call these primitive units such as "(MAKURANO UENIWA)" as "clause"s after English grammar. As we do not make distinction between what they call "phrase" and "clause", we call them just "clause"s.

First, we consider about the size of data to reduce the overheads of control. Some speech understanding systems regard a BUNSETU as a minimal unit to construct a sentence. But, in many cases, we can design the inference subsystem independently of the minimal unit used by the controlling subsystem. As a "clause" is greater than BUNSETUs, the system can predict the conflict of "clauses" in earlier stage of processing. For example, assume that "(TIISANA BEDDONO)" [the small bed] to be a correct "clause". As this "clause" occupies 2 BUNSETUs there left less BUNSETUs to be identified by the speech understanding system. On the other hand, if the system assumes "(BEDDONO)" [bed] to be a correct "clause", the more BUNSETUs left to be identified than previous case. In other point of view, we can consider that an inference and a backtrack are done by more than one BUNSETU at once. Furthermore as number of candidates to be identified reduces, combinatorial explosion would be relieved. According to the above discussion, we adopt the "clause" for the minimal unit of searching. Actually, we classified "clauses" into two classes, "large clause" and "small clause". (Details are described in 3.2).

Next, consider about the efficient search mechanism. In general, a problem solving system does backtracking which often causes loss of the results of inference. Thus the duplicated inference would be done. To archive efficient problem solving, it is important to avoid the duplicated inference. In the ATMS-controlled problem solver, we are provided capability to avoid the the duplicated inference. Therefore, we adopt the ATMS for out speech understanding system.

Fig.2 shows the block diagram of speech understanding system. The whole system consists of three parts, "preprocessor", "structure predictor", "BUNSETU identifier". First, the "preprocessor" extracts clauses from the input. Second, the "structure predictor" predicts possible structures using the extracted clauses. Usually, "structure predictor" predicts more than one structure. Finally, the "BUNSETU identifier" identifies the BUNSETUs referring the predicted structures. We describe the detail of the system below.

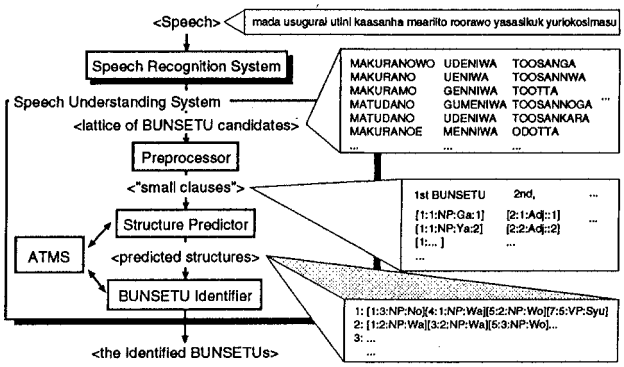

Fig.2: Total System Chart

3.2 Classification of the Clause

Fig.3 shows some examples of clauses. When considering the semantic meaning of the clause, we ignore meanings of modifiers within the clause, because consideration of the modifier meaning brings out a very difficult problem. For example, we con-

sider the meaning of a clause "(YASASIKU YURIOKOSIMASU)" [wake ~ up gracefully] is equivalent to the meaning of a clause "(YURIOKOSIMASU)" [wake ~ up]. In this example, the meaning of YASASIKU is just ignored as the meaning of the clause "(YASASIKU YURIOKOSIMASU)". Of course, we do not ignore the meaning of the word YASASIKU while processing the modification between YASASIKU and YURIOKOSIMASU.

```
TIISANA BEDNO UEKAKENO KIRUTOWO  ---- small noun clause
  [the kilt used as the small bed's bedcover]
PITTARI HAMARUYOUNI KEZTTA KINO  ----- large noun clause
  [... made of wood shaved to fit to the ...]
YASASIKU YURIOKOSIMASU  ---------- small verb clause
  [wake up ... gracefully]
MARUTAWO WARI ---------------- large verb clause
  [cleave the log and ...]
TOTEMO HAYAKU ---------- RENYOUSYUUSYOKU clause
  [very quickly]
YASASIKU ------------ RENYOUSYUUSYOKU clause
  [gracefully]
```

Fig.3: Example of Clauses

We classify clauses into two classes, "small clause" and "large clause". The "large clause" is similar to what they call "phrase". The "small clause" can be classified into some types, "noun clause", "verb clause", "RENYOUSHUUSYOKU clause" and so on. The last one is terminated with adjective or adverb and can modify verb or adjective. A "small noun clause" has no "verb clause" within its component. Like this, a "small verb clause" has no "noun clause" within its component. The "large clause" can be classified into two types, "large verb clause" and "large noun clause". The "large verb clause" is every verb clause but "small verb clause". The "large noun clause" is defined in same way.

3.3 Preprocess

The preprocessor extracts the "small clauses" from an input. As the next module (the predictor) does not require the meaning of the clause to be extracted, this module dose not consider the meanings of clauses. This module extracts the case information only. For example, only the information such as "at the second position, a noun clause whose case is an actor and size is 2 can be assumed." is extracted. Therefore, "small clauses" of the same size and case appears at most once. The extracted "small clause" indicates that there is possibility that such a clause exists.

Preprocessor currently uses the top 20 BUNSETUs in the lattice in which almost syntactic elements appear. This parameter should be modified to fit the speech recognition system's output. An association mechanism[4] is also used to pick up candidates out of the first 20th BUNSETUs. The extracted "small clauses" are scored by preprocessor using the phonetic-score of BUNSETUs which the small clause consists of.

Fig.4 shows some examples of the extracted clauses by preprocessor. The clause is shown as [<position> : <length>(in BUNSETUs) : <case> : <phonemic-score>].

```
[1:1:N:Wo:1]      [3:1:N:Ga:1]
[1:1:N:E:2]       [3:1:N:Wa:2]
[2:1:N:Ga:2]      [3:1:V:Nai/Rentai:3]
[2:1:N:Ha:1]      [4:2:N:Ga:2]
[2:2:N:Ga:2]      [4:2:N:Wa:1]
[2:2:N:Ha:12]     ...

N: Noun Clause
V: Verb Clause
```

Fig.4: Clauses Extracted by the Preprocessor

3.4 Prediction of the Structures

This module is concerned with prediction of the semantic structures using preprocessor's output. This module merely makes clause combination under the condition where the clauses must not overlap each other and checks if it is syntactically consistent or not. This module adds scores to predicted structures. The scores are calculated according to the popularity of the structures themselves and ones given by preprocessor. For example, unpopular structure "... WA ... WA ..." is given less score than popular structure "... WA ... WO ...". †

For this module, the clause's case information is very important. In Japanese, the noun clause's case is given by function words. If a speech recognition system fails to identify the function words, this confronts with a problem. WO and TO, which are function words, are frequently used in a Japanese sentence. Furthermore the word TO is apt to be misrecognized as WO by the speech recognition system. To avoid this problem, the structure predictor consider the possibility of misrecognition TO as WO, only if WO is recognized by the speech recognition system.

At this module, only the case information of clause are used to predict the structure. The semantic information of words and clauses are never referred. Therefore, usually this module predicts more than one structure. For each structure, scores indicating the plausibility of the structures are calculated using the component clause's score and the popularity of the structure itself. These scores are used at the next module to determine which structure should use first.

Fig.5 shows examples of the predicted structures. They are arranged in the descending order of the scores. Each structure is shown as a sequence of clauses each of which is shown in the form similar to the one used in Fig.4.

Rank	Predicted Structures
1.	[[1:2:N:Wa] [3:2:N:Ga] [5:3:N:Wo] [8:1:V:Syu] 980]
2.	[[1:2:N:Wa] [3:2:N:Ga] [5:2:N:Wo] [7:2:V:Syu] 870]
3.	[[1:2:N:Ga] [3:2:N:Wa] [5:3:N:Wo] [8:1:V:Syu] 750]
4.	...
...	

Fig.5: Predicted Structures

3.5 Identification of the BUNSETU

This module identifies the speech. This module consists of two tasks. The first one is to schedule the problem solving. This task decides which structure to be used next. The second one is to identify BUNSETUs consistent with the scheduled structure and the semantic constraints.

† "WA" and "WO" mentioned here are function words.

(1) Scheduling Usually the predictor generates more than one structure. As only one structure among predicted structures leads the correct structure, the system must somehow select a candidate and check if it is correct selection or not. The simplest way to schedule the ordering of structures is sort them by their scores.

(2) Knowledge for Inference Once a structure to be tested is selected, the speech understanding system searches combinations of BUNSETUs, which satisfies the structure and semantic restrictions. This task uses semantics of BUNSETUs and clauses. For example, it examines if "this adjective is able to modify this noun?", "this adverb is able to modify this verb?", "this verb takes this noun as an actor?" and so on.

Some heuristics for accelerating the search are also used at this module. The speech understanding system extracts the topic of the sentence. Then, the search on BUNSETUs which closely relates to one of the topics is performed first. The detail description and evaluation for this heuristics can be found in [4].

(3) Pruning the Search Tree and Avoiding Redundant Inference Task of this module can be understood as a search problem. In general, a problem solving system does backtracking which often causes some problems. First, once backtrack occurs, results of inference is lost. This loss of inference causes redundant inference later. For example, since some predicted structures share the same component, so recording inference history on such components makes it possible to avoid duplicated inferences. Second, if a backtracks occurs frequently, we cannot ignore the overheads about controlling the searching. Using ATMS, we can avoid the first problem easily. The second problem is caused by frequent backtracking. So to resolve this problem, it is important to reduce the number of backtracking. The scheduler uses the "small clause" as the minimal unit to construct the sentence. This module delimits the search area within "small clause"'s boundary. This limitation for search area does not have any influence on problem solver's performance. Therefore, no problem can be brought out by this limitation.

Fig.6 shows the search tree for the structures shown in Fig.5. In the tree, the same node "search on [1:2:N:Wa]" appears twice, and the hatched node is closed within itself. For these reasons, search for the duplicated nodes can be unified to reduce the amount of inferences.

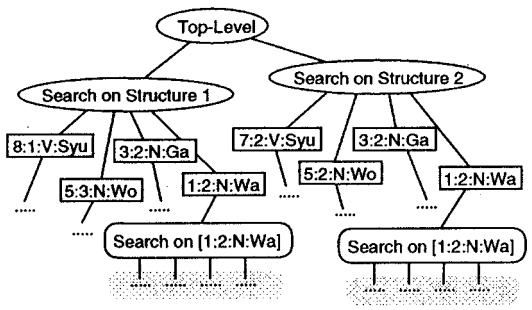

Fig.6: Example of the Search Tree

IV. EVALUATION

We made an experiment to evaluate the framework of the speech understanding system.

The speech understanding system is implemented in Common Lisp. The dictionary of content words used to generate bunsetsus includes 3,000 words in all. The performance of the system is evaluated using 18 utterances. Phoneme sequences simulated with phoneme error rate of 18.6% are given to the system, in the experiment. Length of sentences are 7.4 bunsetsus on an average. There appeared 47.3 candidates on an average in rows of the bunsetsus lattice. When not using our framework, it was hard to measure the system performance, since the system took hours to find solutions. On the other hand, the average time the proposed system took is 3.5 minutes. And the accuracy is 83%.

V. CONCLUSIONS

In this paper, we proposed a new language processing mechanism for the speech understanding systems. We introduced two kinds of phrases such as the "large clause" and the "small clause". The mechanism uses the "small clause" as the minimal unit to construct the sentences. Because the search space is divided into some partitions by this classification, the number of combination of the candidates is reduced. In other words, conventional speech understanding systems must resolve the global and local dependencies simultaneously. Our method deals with them separately, which contributes to efficient language processing.

This classification is useful when the system is under the control of ATMS, too. Because the less the number of the queries to the ATMS, the less overheads. The speech understanding system constructed based on the language processing method and the framework can efficiently identify the input speech.

REFERENCES

[1] J. de Kleer. "An Assumption-based truth maintenance system," Artificial Intelligence, 28, pp.127-162, 1986.

[2] S.Nishioka, M.Hori, M.Ikeda, R.Mizoguchi and O.Kakusho. "Extension of ATMS and its Applications,"(in Japanese), IEICE Tech. Rep. AI88-50, pp.41-50, 1989.

[3] K.Tsujino, R.Mizoguchi and O.Kakusho. "A Continuous Speech Recognition System SPREX: A Knowledge Engineering Approach to Speech Recognition," Proc. of the 3rd Western Pacific Regional Acoustics Conference, Vol.2, pp.771-774, 1988.

[4] M.Hori, K. Tsujino, R.Mizoguchi and O.Kakusho. "A Speech Understanding System SPURT-I: Performance Evaluation with a 1000-word Vocabulary," Proc. of the 3rd Western Pacific Regional Acoustics Conference, Vol.2, pp.779-782, 1988.