Structural Disambiguation in Japanese by Evaluating Case Structures based on Examples in a Case Frame Dictionary

Sadao Kurohashi and Makoto Nagao

Dept. of Electrical Engineering, Kyoto University
Yoshida-honmachi, Sakyo, Kyoto, 606, Japan
email: {kuro|nagao}@kuee.kyoto-u.ac.jp

Abstract

A case structure expression is one of the most important forms to represent the *meaning* of a sentence. Case structure analysis is usually performed by consulting *case frame information* in verb dictionaries and by selecting *a proper case frame* for an input sentence. However, this analysis is very difficult because of *word sense ambiguity* and *structural ambiguity*. A conventional method for solving these problems is to use the method of *selectional restriction*, but this method has a drawback in the semantic marker (SM) system — the trade-off between descriptive power and construction cost.

This paper describes a method of case structure analysis of Japanese sentences which overcomes the drawback in the SM system, concentrating on the structural disambiguation. This method selects a proper case frame for an input by the similarity measure between the input and typical example sentences of each case frame. When there are two or more possible readings for an input because of structural ambiguity, the best reading will be selected by evaluating case structures in each possible reading by the similarity measure with typical example sentences of case frames.

1 Introduction

Representing a sentence with a case structure is a basic form for dealing with its *meaning*. Therefore, transforming a sentence into a case structure expression is one of the most important techniques in natural language processing, and it is needed for machine translation, knowledge acquisition in which various expressions with the same content must be converted into the same representation, and so on.

Case structure analysis is usually performed by consulting *case frame information* in verb dictionaries. The dictionary describes what kind of cases each verb has and what kinds of noun can fill a case slot with what kind of case marker (in Japanese, postpositions (POs) function as case markers). However, this analysis is very difficult because of *word sense ambiguity* (a verb often has two or more meanings and case frames are prepared for the respective meanings) and *structural ambiguity*.

A conventional method for solving these problems is *selectional restriction* (Katz — Fodor, 1963), where the category of the nouns which are able to fill in the case slot is specified by semantic markers (SMs), such as *human*, *animate*, *action*, and so on. However this method has the following weak points.

- The SM system with tens of SMs is too coarse to distinguish every case frame for a verb, which is the case with most electronic dictionaries and systems at present, such as LDOCE (Longman, 1978), the Mu system (Nagao et al., 1985), IPAL (IPA, 1987), and most of commercial machine translation systems.

HAIRU			
Sub-entry 1			
Meaning : Enter from the outside to the inside.			
<*Case markers*>	<*SMs*>	<*Examples*>	<*Deep cases*>
N1-GA	[HUM/ORG/ANI/PRO]	he party cat ship	agent
N2-KARA*	[LOC]	window rear-gate	locational source
N3-NI/E	[LOC]	classroom kitchen port	locational goal / directional
Sub-entry 2			
Meaning: Added to food or drink.			
<*Case markers*>	<*SMs*>	<*Examples*>	<*Deep cases*>
N1-NI	[PRO]	coffee cake	non-locational goal
N2-GA	[CON]	sugar milk cheese poison	object
Sub-entry 3			
Meaning: Be reflected.			
<*Case markers*>	<*SMs*>	<*Examples*>	<*Deep cases*>
N1-NI	[PRO/ABS]	work report proposal	non-locational locative
N2-GA	[MEN]	thought opinion arbitrariness	object

A Case component marked with '*' is optional.

Table 1: Examples of case frames for HAIRU in IPAL.

- On the other hand, it is quite expensive and time consuming to prepare a detailed SM system which has enough descriptive power to discriminate every usage of each verb, and which may require thousands of SMs (Ikehara et al., 1991). A further difficulty is to improve the SM system when needed.

In order to overcome the drawbacks in the SM method — the trade-off between descriptive power and construction cost, we have developed a method of case structure analysis of Japanese sentences based on examples in a case frame dictionary. We published some parts of our analysis system already elsewhere (Kurohashi — Nagao, 1992) (Nagao, 1992). Therefore, this paper concentrates on the structural disambiguation in Japanese complex sentences through the case structure analysis process.

This method uses a case frame dictionary that has some typical example sentences for every case frame. When an input is a simple sentence without structural ambiguity concerning case components, a proper case frame is selected for the verb in the input sentence by matching the input sentence with the examples in the case frame dictionary. When an input is a complex sentence, there would be several verbs to which the nouns in the sentence cannot be linked uniquely by unique case assignment. The important point is that the best matching score, which is utilized for selecting a proper case frame for a verb in a sentence, can be considered as *the score for the case structure* of the verb and its case components. The best reading (the correct reading or the most plausible reading) of a complex sentence is the one where all verbs in the sentence govern appropriate case components and their case structures have high scores. Therefore, the best reading of a sentence can be selected by checking all the possible case structures of all the verbs and by evaluating every possible reading according to the sum of the scores for the case structures in it. When an input is a compound sentence, we can detect the scopes of coordinate structures beforehand, so that we can limit the possible readings to the extent that we can evaluate all of them.

From the viewpoint of an example-based method, there are several research activities for solving structural ambiguity (Inagaki et al., 1988) (Nagao, 1990). Our method has the following characteristics in contrast with them.

- While their methods match an input sentence with examples basically in blocks of two words being in a governor/dependent relation, our method matches them in blocks of a case structure. We can say in general that the wider range of components a method checks, the more reliable it becomes.

- They use texts from their target domain as their main knowledge base. If limiting the text domain, it may be possible and useful, but it is very difficult to cover general domains. On the other hand, we use examples in the case frame dictionary, which can cover wider domains according to entries of the dictionary. Because in compiling a dictionary, lexicographers consult example sentences in which an entry word is used, it is a reasonable assumption that we can use examples in a dictionary in the computer analysis of natural language sentences.

2 Selecting a Proper Case Frame

2.1 Japanese Electronic Dictionaries

In this paper, we use the basic verb dictionary which was constructed by the Information-technology Promotion Agency, Japan (hereafter, this dictionary is referred to as IPAL) (IPA, 1987). In IPAL, 861 basic verbs are entries, and each entry has sub-entries according to the difference in the meaning and the syntax. Case frame information is given at each sub-entry. The sub-entries total up to 3379 so that the average number of sub-entries and thus the average number of case frames for a verb is 3.9. As shown in Table 1, a set of case frame information consists of the meaning of a verb, its case markers, SMs, examples and correspondences to deep cases for each case slot. SMs restrict the category of the nouns that can fill in the case slots. IPAL uses 19 different SMs which have the tree structure shown in Figure 1. However, our method does not use this SM system because of its coarseness. We use a thesaurus dictionary, 'Bunrui Goi Hyou' (abbreviated as BGH) (NLRI, 1964) for calculating the similarity values between words. BGH has a tree of six layer abstraction hierarchy and more than 60,000 words are assigned to the leaves of the thesaurus tree.

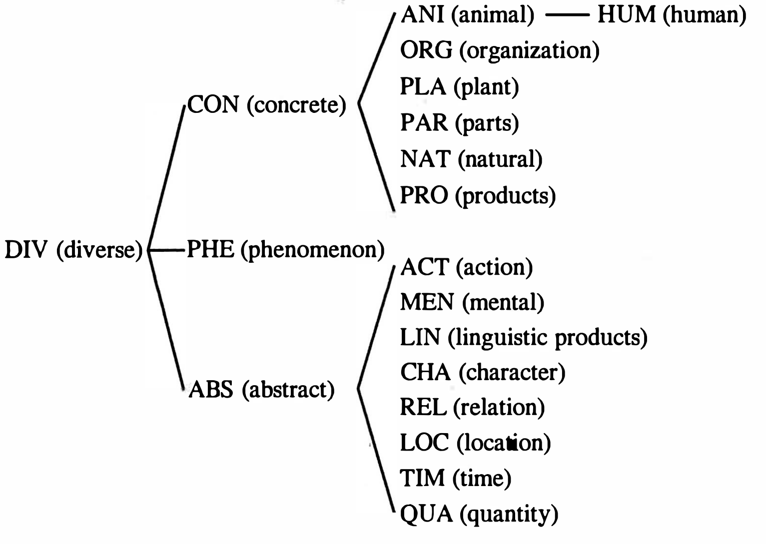


Figure 1: The set of SMs in IPAL.

2.2 Case Structure Analysis by Case Frames

Case structure analysis is usually performed by consulting case frames in a dictionary. Because the correspondence of each case component to a deep case is listed in the case frame, we can get deep cases for case components when we find a proper case frame for an input sentence and the correspondence of case components in the sentence to those in the case frame. In other words, case structure analysis is regarded as a selection of a proper case frame for an input sentence.

2.3 Selecting a Proper Case Frame

A conventional method for selecting a proper case frame is selectional restriction by SMs. However, the SM system with tens of SMs, such as IPAL, is too coarse to select a proper case frame for an input sentence. For instance, in selecting a proper case frame for the example sentence (ES1):

KISHA-GA	TON'NERU-NI	HAIRU.
(train)	(tunnel)	(enter)
[PRO]	[LOC/PRO]	

out of the case frames in Table 1, while sub-entry 3 can be removed by comparing the SM [PRO(product)] of 'KISHA(train)' with the SM [MEN(mental)] of case slot 'N2-GA', the incorrect case frame, sub-entry 2, is selected together with the correct case frame, sub-entry 1. On the other hand, it is quite expensive and time consuming to prepare a detailed SM system which has enough descriptive power to discriminate every usage of each verb.

In order to overcome this drawback in the SM method, we have developed a method for case structure analysis of Japanese sentences based on examples in a case frame dictionary. In brief, this method selects the case frame whose example is the most similar to the input sentence. Because ES1 above is much more similar to examples of sub-entry 1, "(he party cat ship)-GA (classroom kitchen port)-NI", than to examples of sub-entry 2, "(coffee cake)-NI (sugar milk cheese poison)-GA", the correct case frame, sub-entry 1, can be selected.

The similarity score between an input sentence and examples of a case frame is calculated by the following algorithm. The algorithm assumes that case components depending on a verb are already known, as in the case of processing a simple sentence.

1. **Matching case components.**
   First, case components of the input sentence and those of a case frame are matched by the equality of POs. A noun modified by a clause sometimes becomes a case component for the verb of the modifying clause. In this case, the modified noun can correspond to case slots followed by PO 'GA', 'WO', 'NI' or 'DE'.

2. **Calculating the score of a matching case component.**
   A score of matching case components is defined as the greatest similarity value between a noun of the input sentence and example nouns assigned to a case slot in the case frame dictionary. The similarity value (SV) between two nouns is given according to the most specific common layer (CL) between them in BGH, as follows:[1]

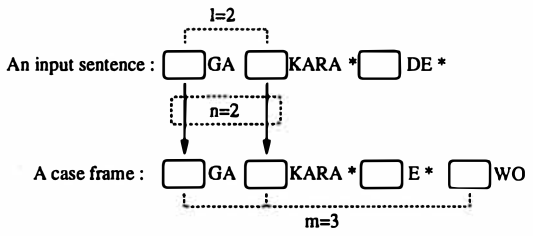
CL	0	1	2	3	4	5	6	exact match
SV	0	0	5	7	8	9	10	11

3. **Calculating the score of a matching pattern.**
   It is assumed that *the matching pattern* between an input sentence and a case frame is given as follows:

[1] This way of correlating the most specific common layer (CL) with the similarity value has the following basis ($sv(i)$ means the similarity value between two nouns whose CL is i).

(a) Since the first layer of BGH consists of four classes: nouns, verbs, adjectives and the others, sharing the first layer of two nouns does not indicate that they are similar. Therefore, we let $sv(1)$ '0'.

(b) The greater the CL between two nouns is, the more similar they are. Furthermore, by studying BGH, we concluded that sharing the general layer (except the first layer) has more effect on the similarity between two nouns than sharing the specific layer. For this reason, $sv(i)$ is designed to simulate a convex and monotone increasing function.



(* is attached to optional components)

n : the number of matching components.

l : the number of matching or obligatory case components in the input sentence. Case components followed by PO 'GA', 'WO', 'NI', 'E' or 'YORI', are regarded as obligatory case components.

m : the number of matching or obligatory case slots in the case frame. Obligatory case slots are specified in IPAL.

total_score : the sum of scores of matching case components.

The simplest way is to regard the *total_score* as the score of this matching pattern. However, we need to take more factors into consideration. We give the following score to this matching pattern:

$$\begin{cases} \text{if } l > n & 0 \\ \text{otherwise} & total\_score \times \left(\frac{1}{n}\right)^{1/2} \times \left(\frac{n}{m}\right)^{1/2} \end{cases}$$

We let the score '0' when l is greater than n, because the obligatory case components cannot remain unmatched in the sentence for its proper case frame. We include $(n/m)^{1/2}$ in the above formula in order to give priority to case frames with the higher ratio of matching case components to the total number of case components. We also include $(1/n)^{1/2}$ because it is preferable not only that there are many matching components but also that the scores of matching components are big. The exponents of $(n/m)^{1/2}$ and $(1/n)^{1/2}$ were determined empirically.

Matching calculation is performed for all the matching patterns between the input sentence and all case frames, and then the case frame whose matching pattern has the greatest score is selected as the final result.

The experiment of comparing the example based method with the SM method in IPAL, and the discussion about the validity of the example based method were reported in (Nagao, 1992).

3 Structural Disambiguation using Case Structure Score

3.1 Outline of the Method

In the preceding section we described a method of analyzing the case structure for an input sentence when case components depending on a verb are already known, as in the case of a simple sentence. This section introduces the way of extending the method to process complex or compound sentences.

Japanese sentences can best be explained by "Kakari-uke", which is essentially the governor/dependent relation between *bunsetsus*.[2] A bunsetsu depends on, that is, modifies another bunsetsu to its right (not necessarily the adjacent bunsetsu). Sometimes a bunsetsu can depend on two or more bunsetsus, which creates structural ambiguity and makes case structure analysis hard. Other work concerning structural disambiguation (Inagaki et al., 1988) (Nagao, 1990) solves this problem locally, that is, they try to determine the governor of each bunsetsu independently. However, in order to improve the precision of analyzing sentences, the ambiguity of the governor of a bunsetsu must be processed simultaneously with the ambiguity of the governors of other bunsetsus and with the word sense ambiguity.

[2] Bunsetsu is the smallest meaningful block consisting of an independent word (nouns, verbs, adjectives, etc.) and accompanying words (POs, auxiliary verbs, etc.).

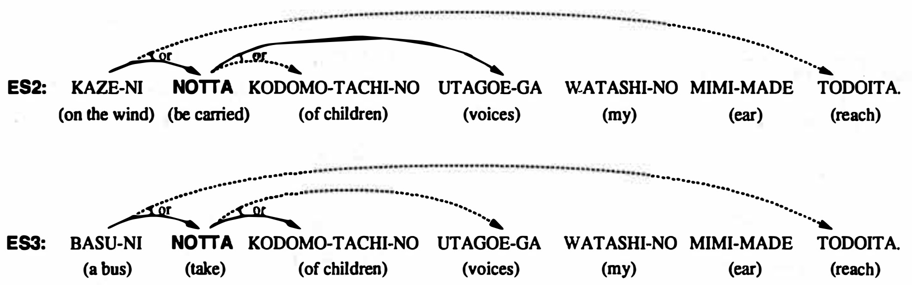


Figure 2: Need for a global disambiguation.

The ambiguity in example sentences in Figure 2 makes this problem clear. In ES2, the meaning of the verb 'NOTTA' is 'be carried (on the wind)', and 'KAZE-NI(on the wind)' depends on 'NOTTA(be carried)' and 'NOTTA(be carried)' depends on 'UTAGOE-GA(voices)'; whereas, in ES3, the meaning of the same verb 'NOTTA' is 'take (a bus)', and 'BASU-NI(a bus)' depends on 'NOTTA(take)' and 'NOTTA(take)' depends on 'KODOMO-TACHI-NO(of children)'. This means that whether 'NOTTA' depends on 'KODOMO-TACHI-NO(of children)' or 'UTAGOE-GA(voices)' can not be determined independently of the structural ambiguity of other case components of the verb 'NOTTA' and its word sense ambiguity.

For such a global disambiguation, we can use the best matching score which is utilized for selecting a proper case frame for each verb. The best matching score between an input sentence and a typical example for the usage of a verb can be considered as *the appropriateness* (score) for the case structure of the verb and its case components. When there are two or more readings (dependency structures) for a sentence because of structural ambiguity, the best reading (the correct reading or the most plausible reading) is the one where all verbs in the sentence govern appropriate case components and their case structures have high scores. This means that the best reading of a sentence can be selected by evaluating the sum of the scores for the case structures of all verbs in a sentence (Figure 3).

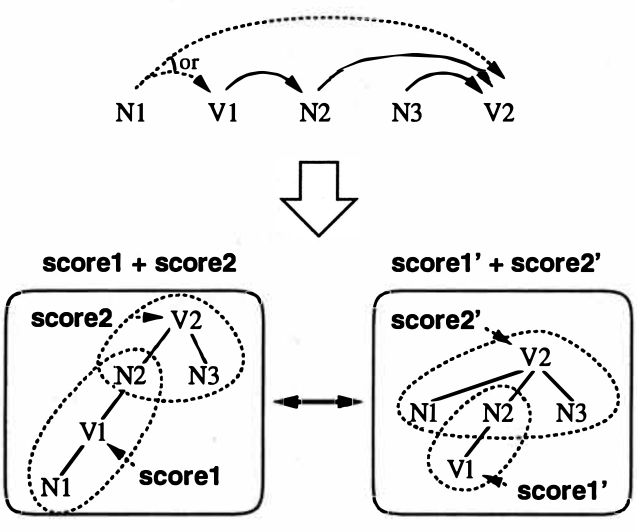


Figure 3: Outline of the method.

1	**Key bunsetsu of coordinate structure (KB)**	Bunsetsu which indicates the existence of a coordinate structure, such as "...SHI", and "...TO". This type is treated as depending on the last bunsetsu in the coordinate structure.
2	**PB depending on PB**	PB in a kind of subordinate structure, such as "...SURE-BA (if...V)", "...SITA-NODE (because ... V)".
3	**NB depending on PB**	Case components, such as "...GA", "...WO".
4	**PB depending on NB**	PB in a clausal modifier, such as "...SITA (which ... =V)". A governor of this type of a bunsetsu may become a case component for the bunsetsu.
5	**NB depending on NB**	NB, such as "A-NO" in the noun phrase "A-NO B".

Table 2: Dependent types.

In order to evaluate all the possible readings, it is necessary to expand all the structural ambiguity for a sentence. However, before entering this analysis stage, we detect the scope of coordinate structures in the sentence by using another method (Kurohashi — Nagao, 1992) to avoid the combinatorial explosion problem. The main reason that a sentence becomes long, particularly in Japanese, is that two or more matters are expressed in a sentence, that is, a sentence has coordinate structures. Therefore, by detecting the scopes of coordinate structures beforehand, the possible readings are limited to the extent that we can evaluate all of them.

We will explain this method in detail in the following subsections.

3.2 Calculation of the Possible Dependency Matrix (PDM)

For evaluating each possible dependency structure, we first get all the possible governor/dependent relations between two bunsetsus. These relations are expressed in the form of a triangular matrix $A = (a_{ij})$ (Figure 4), called possible dependency matrix (PDM), whose diagonal element a_{ii} is the i-th bunsetsu (hereafter expressed as B_i) in a sentence and whose element a_{ij} (i < j) expresses whether B_i can depend on B_j ('1' means yes).

As a governor, each bunsetsu is classified into one of two types according to parts of speech; nominal bunsetsu (abbreviated as NB) and predicative bunsetsu (abbreviated as PB). As a dependent, each bunsetsu is classified into one of five types in Table 2 according to its PO or conjugation.

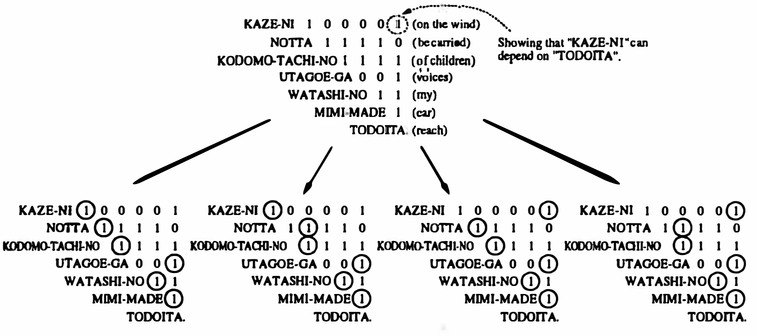


Figure 4: Making the possible dependency structures by consulting the PDM (ES2).

The way of determining an element a_{ij} is rather simple. The value of element a_{ij} is adjusted to '1' when B_i is type 2 or type 3 (which can depend on PB) and B_j is a PB, or when B_i is type 4 or type 5 (which can depend on NB) and B_j is a NB.

3.3 Masking the PDM by the Scope of the Coordinate Structure

When an input sentence contains a key bunsetsu of coordinate structure (abbreviated as KB) which indicates the existence of a coordinate structure, we detect its scope by the method in which the two most similar series of bunsetsus on the left and right side of the KB are detected and are regarded as the scope of the coordinate structure concerning the KB (see (Kurohashi — Nagao, 1992) for details).

After detecting the scopes of coordinate structures in a sentence, the following two operations are performed on the PDM (Figure 5).

- *Setting the governor of a KB*: A KB is treated as depending on the last bunsetsu of a coordinate structure. When B_i is a KB and B_j is the last bunsetsu of its scope, the value of the PDM element a_{ij} is adjusted to '1'.
- *Masking the PDM*: Because the prior and the posterior parts of a coordinate structure have their own consistent structures and meanings, they are parsed independently into dependency structures. Therefore, the bunsetsu in a coordinate structure does not depend on or become governor of any bunsetsu outside its scope, except the last bunsetsu of the coordinate structure which has governor/dependent relations to bunsetsus outside its scope. In order to express these characteristics, the value of the PDM elements on the upper and right side of a coordinate structure are set to '0'.

As a result of this process, the number of possible dependency structures of a sentence can be reduced drastically.

3.4 Making the Possible Dependency Structures

Next, we expand the ambiguities of the case components of an input sentence and make the possible dependency structures by consulting the PDM. Because governor/dependent relations do not cross each other in Japanese, *no-cross condition* can be set as follows: when B_i depends on B_j, B_k (k < i) cannot depend on bunsetsus from B_{i+1} to B_{j-1}. For the following explanation, we

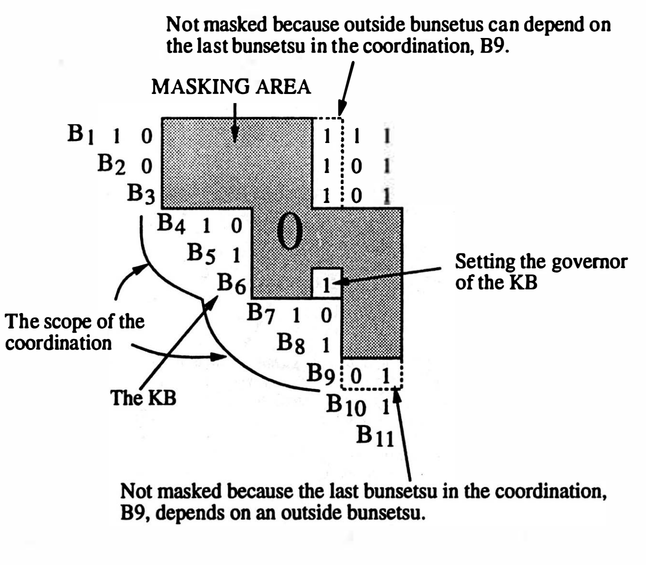


Figure 5: Masking the PDM.

define *a dependency set* as a set of a bunsetsu consisting of bunsetsus on which the bunsetsu can depend. This set is fixed dynamically by the PDM and the no-cross condition.

The governor is determined for each bunsetsu from right to left. When a bunsetsu concerning a case component (type 3 or type 4 in Table 2) has the possibility of depending on two or more bunsetsus (that is, its dependency set consists of two or more bunsetsus), two or more partial dependency structures are made according to the varieties of its governor, and the structures on the left side of each partial structure are analyzed. On the other hand, when a bunsetsu, not concerning case components (type 2 or type 5), can depend on two or more bunsetsus, its governor is determined uniquely to be the nearest bunsetsu in its dependency set, because a bunsetsu usually depends on the nearest bunsetsu in Japanese (of course, this heuristic rule sometimes makes a mistake. We will deal with this problem in future work).

In the case of ES2 (Figure 4 on page 117), because 'NOTTA (be carried)' can depend on either 'KODOMO-TACHI-NO (of children)' or 'UTAGOE-GA (voices)' and 'KAZE-NI (on the wind)' can depend on either 'NOTTA (be carried)' or 'TODOITA (reach)', four possible dependency structures are created.

3.5 Evaluation of Possible Dependency Structures

Case frame selection is performed for all verbs in each possible dependency structure which is made in the above-mentioned processes. Of all possible dependency structures, we select the structure which has the maximum sum of the best matching scores for all verbs in the sentence. In the case of ES2, the correct structure, the second one form the left in Figure 4, is selected by this method, selecting proper case frames for verbs 'NOTTA' and 'TODOITA' (the sub-entry whose meaning is "be carried" is selected for 'NOTTA' correctly).

If there are two or more structures which have the maximum score, the structure which is most similar to the *default dependency structure* is selected. Here, a default dependency structure is that in which each bunsetsu depends on its nearest bunsetsu in its dependency set.

Our method	MT systems	Type3	Type4	Total
○	○ ×	56	12	68
○	× ×	23	15	38
×	○○	5	0	5
×	○×	6	0	6
○	○○	104	6	110
×	× ×	2	4	6

Table 3: Comparison between our method and commercial MT systems.

4 Experiment

We report a experiment which illustrates the effectiveness of this method for solving structural ambiguity. This method limits the possible readings by detecting coordinate structures beforehand; the validity of the method for detecting coordinate structures has already been reported in (Kurohashi — Nagao, 1992). After detecting coordinate structures, the remaining problem is the ambiguity in a complex sentence. Therefore, in this paper, we show an experiment of analyzing complex sentences.

We had a language-trained person compose a set of about 450 complex test sentences each of which includes one or more clausal modifier. Then we analyzed these test sentences by our method and evaluated the analysis results from the viewpoint of structural disambiguation according to the following structural types of sen-

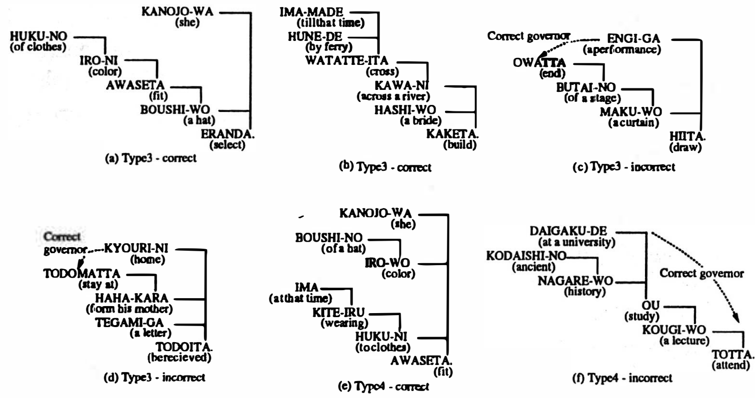


Figure 6: Examples of detecting dependency structures.

tences.

Type 1 : A sentence which has no structural ambiguity concerning case components.

Type 2 : A sentence which has two or more correct dependency structures.

Type 3 : A sentence which has two or more possible dependency structures and whose correct dependency structure is its default dependency structure.

Type 4 : A sentence which has two or more possible dependency structures and whose correct dependency structure is not its default dependency structure.

The left part of Table 4 shows its results. This table shows that the success ratio of getting a correct dependency structure by this method is very high. Examples of correct and incorrect analysis are shown in Figure 6. The reasons of incorrect analysis are listed below.

- Inadequacy of the case frame dictionary IPAL regarding surface cases, distinction between obligatory case and optional case, and category specification for case slots (which means oversight of examples).
- Inadequacy of the thesaurus dictionary BGH. This problem is closely related to the method for correlating the level of most specific common layer with their similarity value. Generally speaking, however, BGH is not reliable enough to calculate an accurate similarity value between words.
- Insufficiency of examples in the case frame dictionary. Some case slots have only one or two examples. This problem can be solved simply by adding the wrongly analyzed sentences as new examples of their proper case frames.

In order to see how well these test sentences are analyzed by conventional SM methods, we translated test sentences of type 3 and 4, which have structural ambiguity, by two commercial machine translation (MT) systems. The commercial MT systems have a lot of heuristic rules, but they are thought to be based on tens or hundreds of SMs. We evaluated their outputs based on whether the syntactic analysis of Japanese sentences is correct or not (in the right part of Table 4). Furthermore, we compared the analysis results by our method with those by commercial MT systems (Table 3). We can see that the disambiguation of such complex sentences are fairly difficult for conventional SM methods and that our example-based method is significantly better at structural disambiguation than conventional SM methods.

	Our method			MT system I			MT system II		
	correct	incorrect	sr*	correct	incorrect	sr*	correct	incorrect	sr*
Type 1	219	0	100%	-	-	-	-	-	-
Type 2	12	0	100%	-	-	-	-	-	-
Type 3	183	13	93%	133	63	68%	147	49	75%
Type 4	33	4	89%	12	25	32%	12	25	32%
Type 3 and 4	216	17	93%	145	88	62%	159	74	68%

*success ratio

Table 4: Results of experiments.

5 Concluding Remarks

We have proposed a method that detects the case structure not only for a simple sentence, but also for a compound or a complex sentence. In this method, word ambiguity for verbs and structural ambiguity are solved simultaneously. The basis of this method is the process of selecting a proper case frame for the input sentence by matching it with example sentences in the case frame dictionary. We have reported experiments showing this method's superiority over the conventional, coarse-grained SM method.

The remaining problems are:

- In this paper we have hardly discussed the concept of thesaurus. Not only case structure analysis but also many other kinds of natural language processing depend on the accuracy of the thesaurus employed. We need to do research on the framework of a thesaurus where the relations of words are handled in various aspects and also research on a method for automatic construction of such a thesaurus.

- At present we first detect the scope of the coordinate structure and then detect the case structure of a sentence. However, it is desirable that the coordinate structure and the case structure of a sentence are evaluated by one combined measure as a whole. In order to do this without the combinatorial explosion of ambiguities, we need to devise a data structure and a search method for handling these problems together, or need to devise a method for judging dynamically which information is the most reliable.

References

[1] Katz, J. — Fodor, J. (1963) "The structure of a semantic theory". In: *Language* 39, 170-210.

[2] Longman Group Ltd. (1987) *Longman Dictionary of Contemporary English.*

[3] Nagao, M et al. (1985) "Outline of Machine Translation Project of the Science and Technology Agency". In: *J.IPS Japan* Vol.26, No.10 (in Japanese).

[4] Information-technology Promotion Agency, Japan (1987) *IPA Lexicon of the Japanese Language for computers IPAL (Basic Verbs).* (in Japanese).

[5] Ikehara, S et al. (1991) "Semantic Analysis Dictionaries for Machine Translation". In: *IPSJ-NLP* 84-13 (in Japanese).

[6] Kurohashi, S — Nagao, M (1992) "Dynamic Programming Method for Analyzing Conjunctive Structures in Japanese". In: *Proc. of the 14th International Conference on Computational Linguistics.*

[7] Nagao, M (1992) "Some Rationales and Methodologies for Example-based Approach". In: *Proc. of Workshop on Future Generation Natural Language Processing.* UMIST, Manchester.

[8] Inagaki, H et al. (1988) "Modification Analysis using Semantic Pattern". In: *IPSJ-NLP* 67-5 (in Japanese).

[9] Nagao, K. (1990) "Dependency Analyzer : A Knowledge-Based Approach to Structural Disambiguation". In: *Proc. of the 13th International Conference on Computational Linguistics.*

[10] National Language Research Institute (1964) *Word List by Semantic Principles.* Syuei Syuppan (in Japanese).