[TMI 99: *Proceedings of 8th International Conference on Theoretical and Methodological Issues in Machine Translation*, August 1999, Chester, UK]

Automatic Generation of Semantic Dependency Rules for Japanese Noun Phrases with Particles "no"

Satoru Ikehara, Shinnji Nakai, Jin'ichi Murakami
* *Tottori University, Koyama, Tottori-city, Japan*
{ikehara, nakai, murakami}@ike.tottori-u.ac.jp

Abstract

The most important problems of dependency analysis are separated into three kinds of dependencies: noun to verb dependency, verb to verb dependency and noun to noun dependency. Recently, efficient methods have been developed for the first two problems. However, the third problem, namely noun phrase analysis, has not yet been solved.

One of the most typical and important noun phrases in the Japanese language is the expression “AのBのC” (similar to “A of B of C” in English), which is composed of three nouns, A, B and C. In order to analyze this type of noun phrase, this paper proposes a method to generate dependency rules automatically. In this method, semantic structures of noun phrases are defined using *a semantic attribute system* which has recently been developed by NTT Japan. 4 kinds of semantic dependency rules, *one-dimensional, two-dimensional, three-dimensional rules* and *exceptional rules* were generated in the order of generality using examples extracted from the corpus.

In the experiments, this method was applied to 10,000 Japanese noun phrases; 198 rules, 1,480 rules, 136 rules and 0 rules, respectively, were generated for each of the above 4 rule groups. The generated rules were applied to the dependency analysis, resulting in an accuracy rate of 85.8%.

1. Introduction

One of the most important problems in natural language processing is syntactic and semantic ambiguities of expressions. Syntactic ambiguities of noun phrases, compound nouns and the conjugative relation of predicates are considered to be the most difficult problems in Japanese sentence analysis. On the other hand, in the case of semantic analysis, the most important problem is how to determine the meaning of words, especially nouns and verbs. There have been many studies (Kurohashi & Nagao 1992, Shirai et al. 1995) concerning these problems. However, it has proven difficult to get good results without using semantic information.

Currently, a semantic dictionary, *"GoiTaikei-A Japanese Lexicon",* has been developed (Ikehara 1987, Ikehara et al. 1993) and published by Iwanami Shoten, Japan (Ikehara et al. 1997). In this dictionary, semantic uses of 400,000 Japanese words are defined using a semantic attribute system (2,700 categories). Also, the 14,000 case frame pairs of Japanese to English are defined for 6,000 Japanese verbs using semantic attributes. This dictionary has made it possible to make correct correspondence between the meanings of Japanese verbs and English verbs. This dictionary can also be used to determine dependency relationships between verbs and nouns (Ikehara et al. 1993). However, compared to the relationships between verbs and nouns, relationships between two or more nouns are more complicated, and there is no good method for noun phrase analysis.

Fortunately, it is not beyond expectation to think that this dictionary can be applied to the analysis of noun to noun relationships, because the semantic uses of a huge number of nouns are precisely defined. Next, based on the semantic information defined in this dictionary, this paper proposes a method to generate automatically dependency rules between nouns in Japanese noun phrases with the particle *"no"* and evaluates the quality of the generated rules.

As for analysis of Japanese noun phrases, there are many researches, such as those to decide dependency relationships using word co-occurrence information (Sasaki et al. 1995), the method to obtain dependency probability between nouns in compound words using semantic classes of words (Kobayashi 1996) and example-based translations for noun phrases from Japanese to English (Sumita & Iida 1992). These conventional approaches were based on the statistical method (Nagao 1984, Sato 1992), which requires a huge number of examples. However, examples in corpus are usually very limited and yield only sparse data so that it is difficult to get sufficient dependency rules.

In order to solve this problem, this paper uses semantic attributes instead of noun words. In this method, first, noun phrase examples are collected from the corpus. Then, after a person determines the dependency relationship, each noun in the examples is replaced by its semantic attributes by referring to *the Japanese Lexicon.* This procedure yields the semantic patterns for each noun phrase example. Then, these patterns are statistically analyzed to obtain generalized dependency rules in the order of generality. The obtained rules are then applied to the dependency analysis of the noun phrases in the actual documents to evaluate their quality.

2. Semantic Structure of Expressions

2.1 Meanings of Noun Phrases

(1) Ambiguity of the Noun Phrases

Let us consider one of the most popular Japanese noun phrases, “A *no* B *no* C,” which is composed of two *"no"* particles and three nouns, A, B and C. This kind of noun phrase is similar to the English noun phrase, “A of B of C”. Here, let’s consider the dependency relationship between the nouns A, B and C. In Japanese sentences, any word modifies only a word following it. It does not modify a word preceding it. This is known as *the principle of forward modification* in Japanese. The noun B always modifies the noun C in this type of noun phrase, so that there are only two interpretations for the dependency relationship of noun A, as shown in the following:

1) Case of A → B (& B → C) : *b-dependency*

Ex.) Reading: *watasi-no haha-no namae*
Noun phrase: 私の 母の 名前 (name of my mother)

Ex.) Reading: *yokushitsu- no datsuijou- no kabe*
Noun phrase: 浴室の 脱衣場の 壁 (wall of the dressing room of the bath)

2) Case of A → C (& B → C) : *c-dependency*

Ex.) Reading: *watasi-no mukasi-no tomodachi*
Noun phrase: 私の 昔の 友達 (my old friend)

Ex.) Reading: *Tokyo-no suugaku-no kyoushi*
Noun phrase: 東京の 数学の 教師 (teacher of math in Tokyo)

Here, $\alpha \rightarrow \beta$ represents the dependency relationship in which the noun a modifies the noun β , and α and β are known simply as *the dependent* (or *modifier)* and *the head* respectively. For brevity, the dependency relationships of 1) and 2) are simply called *b-dependency* and *c-dependency* respectively.

(2) Semantic Distance of Nouns

One way of analyzing the dependency relationship between two nouns is to use the distance between their meanings. The semantic distance between any two nouns can be calculated from the thesaurus, which is constructed based on inclusion relationships *(is-a relationship)* and part-to-whole *(has-a relationship).* In this method, priority will be given to the dependency relationship between the nouns which have a shorter distance. Let's look at the following examples:

(a) Example of *c-dependency*
Reading: *yakyuubu-no mae-no kantoku*
Noun phrase: 野球部の 前の 監督 (the previous manager of the baseball team)

(b) Example of *b-dependency*
Reading: *zou-no hana-no saki*
Noun phrase: 象の 鼻の 先 (tip of an elephant's nose)

In example (a), the thesaurus will give *a has-a relationship* between the noun A (野球部 : baseball team) and the noun C (監督: manager); thus the dependency relationship A → C *(c-dependency)* will be selected. In the case of (b), the thesaurus will give *a has-a relationship* between the noun A (象: elephant) and the noun B (鼻: nose); thus the dependency relationship A → B *(b-dependency)* will be selected.

However, as shown in (c) and (d), there are many case in which the thesaurus does not give sufficient information to decide the dependency relationship:

(c) Example of *c-dependency*
Reading: *watasi-no kiniro-no nekkuresu*
Noun phrase: 私の 金色の ネックレス(my gold-colored necklace)

(d) Example of *b-dependency*
Reading: *yamazato-no fuyu-no sabisisa*
Noun phrase: 山里の 冬の 寂しさ (loneliness of winter in a mountain village)

There are many kinds of semantic relationships for the noun phrase, "A *no* B". These have already been studied and classified into about 80 categories (Shimazu et al. 1986, 1987, Tomiura et al. 1995) from the view point of linguistics. If we were able to use them, we would be able to select a suitable pair from the two noun phrases, "A *no* B" and "A *no* C". However, it is difficult to decide the meanings by computer.

(3) Structural Meanings of Expressions

If we consider the relationship between the meanings and the structures of expressions, we will find that the method that separates the noun phrase into "A *no* B" and "A *no* C" does not always work well. According to *the principle of compositional semantics,* the meanings of expressions can be considered as a function of their elements. However, there are many cases in which this principle does not hold in natural languages. In such cases, the structural meanings of expressions need to be considered. For example, the next noun phrase is not separated into "A *no*", "B *no*" and "C", because the idiomatic expression of "B *no* C" cannot be separated without losing the meanings.

Readings:	nusutto-no	nare-no	hate
Japanese:	盗人の	なれの	果て (ruined state of a thief)
English :	thief, of,	custom, of,	end

In this case, "B *no* C" needs to be processed as a whole, not separated into parts. Thus, also, in the generation of dependency rules for noun phrases, it is important to distinguish the expressions that cannot be separated into parts without losing their meanings from those that can be separated.

2.2 The use of Semantic Attributes

(1) Description of Semantic Structure

Let us consider how to describe the semantic structure of noun phrases to define dependency rules. Here, we assume that the noun phrases have no more modifiers and the meaning is determined independently from the context. Then, the dependency relationships between three nouns, A, B and C can be uniquely determined. If we can collect examples from actual documents which include every pair of individual nouns, these examples can be used as a rule to determine the dependency relationship.

But it is not possible to collect sufficient examples for every combination of three nouns. In order to obtain generalized dependency rules from examples, one must replace each noun in the examples by some kind of symbol, such as semantic marker or semantic features. But, conventional systems of these symbols (Mizutani et al. 1983, Pollard et al. 1994, Matsumoto et al. 1997) are too coarse to define the semantic relationship without losing the meaning.

Fortunately, the semantic dictionary, *the Japanese Lexicon,* has recently been developed. In this dictionary, semantic uses of a huge number of words (400,000 words) were defined by using a minute semantic attribute system (2,700 categories). It can be expected that the semantic structures of noun phrases and their dependency relationships can be defined correctly using this system.

(2) Outline of Semantic Attribute System

The Japanese Lexicon is comprised of the following: *Semantic Attribute System, Semantic Word Dictionary* and *Semantic Structure Dictionary for verbs.* Characteristics of *the Semantic Attribute System* are as follows:

a) General Noun Semantic Attribute System

This system defines the names of the semantic attributes of general nouns and their relationships. As shown in Fig. 1, it has a tree structure with 2,700 categories and has 12 levels in depth. Links from upper nodes to lower nodes represent *is-a relationship* or *has-a relationship.*

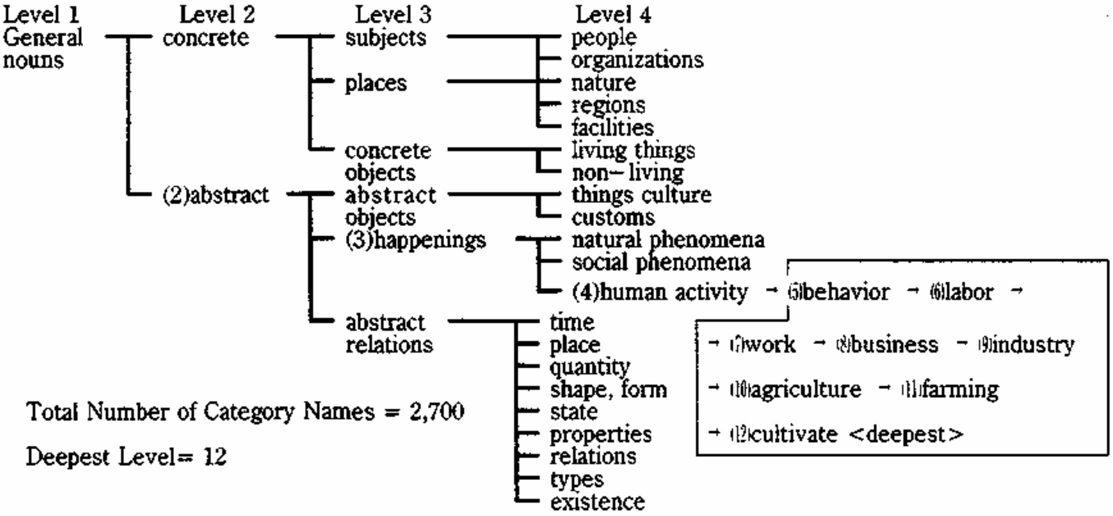


Fig. 1 General Nouns Attribute System (Upper 4 levels)

b) Proper Noun Semantic Attribute System

The proper noun part of the above system is more minutely re-defined in this system. It has a tree structure with 130 categories and depth of 9 levels.

In *the Semantic Word Dictionary,* semantic use of 400,000 Japanese nouns are defined by using the above *Semantic Attributes.*

3. Dependency Rule Generation

3.1 Format and Types of Rules

(1) Format of Dependency Rules

Replacing the nouns A, B and C by their semantic attributes, semantic structures of the noun phrases are obtained and dependency rules can be defined for each of them. Let us represent the semantic structure of the noun phrase of "A *no* B *no* C" by (X, Y, Z). Here, X, Y and Z represent the i. d. -numbers of the semantic attributes for the nouns A, B and C. The dependency rules for the semantic structures (X, Y, Z) are denoted by (X, Y, Z: D) as shown in Fig. 2.

(X, Y, Z: D): General form of dependency rules

X: *a semantic attribute number of noun* A, Y: *a semantic attribute number of noun* B
Z: *a semantic attribute number of noun* C
D: *type of dependency relations (b: b-dependency, c: c-dependency)*

Fig. 2 Format of Dependency Rules

Here, note that D means the dependency relationship, *b-dependency* or *c-dependency.* When the attribute at the noun A position is defined as X in a rule, it means that the rule can be applied to noun phrases in which the attribute of noun A belongs to node X or to nodes lower than X that are connected to node X in *the Semantic Attribute System.*

(2) Types of Dependency Rules

Although nouns A, B and C were replaced by their semantic attributes, it turns out to be nearly impossible to define the dependency relationships for every semantic structure because of the number, which amounts to almost $2,700^3$. In order to reduce the rules for actual use, further generalization will be needed. We can see the same problems in the case of *case frame learning* (Utsuro et al. 1993, Almuiallim et al. 1994). But, conventional methods could not give an accurate solution.

In this paper, we assumed that if the semantic attributes of the three nouns are given, most of the dependency relationships can be uniquely determined. However, there will be many cases in which all of the attributes of the three nouns are not necessarily required to determine the dependency relationships. Next, we separate the rules into the following 4 groups:

1) One-dimensional Rules

This is the rule group defined by the semantic attributes of a single noun. In this case, if one of the three semantic attributes is given, the dependency relationships are uniquely determined independently from the other two nouns. Then, this group is further classified into 3 groups based on the noun position, which has semantic constraints.

2) Two-dimensional Rules

This is the rule group defined by the semantic attributes of two nouns. In this case, if two of the three semantic attributes are given, the dependency relationships are uniquely determined independently from the other noun. This group is further classified into 3 groups based on the noun positions, which have semantic constraints.

3) Three-dimensional Rules

This is the rule group defined by the semantic attributes of all of three nouns.

4) Exceptional Rules

This is the group for which dependency relationships cannot be determined by the relations of semantic attributes. In this case, dependency rules are defined using word face for each noun.

In the definition of dependency rules, semantic attribute number “0” represents the root node, which includes all of the nouns. This means that no constraint exists for the noun represented by the number “0” in the dependency rules. Thus, the rule’s forms for the above 4 groups will be as follows:

1) Format of one-dimensional rules	(X, 0, 0: D), (0 ,Y , 0: D), (0, 0, Z: D)
2) Format of two-dimensional rules	(X, Y, 0: D), (0, Y, Z: D), (X, 0, Z: D)
3) Format of three-dimensional rules	(X, Y, Z: D)
4) Format of exceptional rules	(A, B, C: D)

(3) Order of Rule Generation

Assuming that the rules are applied in the order of generation when applying the generated rules to dependency analysis, the rules are generated in the order of lower dimensional rules to higher dimensional rules. And every time that rules are extracted from the noun phrase data, the data used for the rule generation are deleted. For example, after *one-dimensional rules* are extracted, the data used for them are deleted from the learning data set and *two-dimensional rule sets* are extracted from the remainder. In this way, the total number of rules will be reduced without decreasing the accuracy of the rules.

3.2 Rule Generalization

(1) ***One-dimensional Rule*** **Generation**

Prepare the trees of *the Semantic Attribute System* for each noun A, B and C. Give value sets (m_i, n_i) to each node of these trees. The example of the case of the first noun A is shown in Fig. 3.

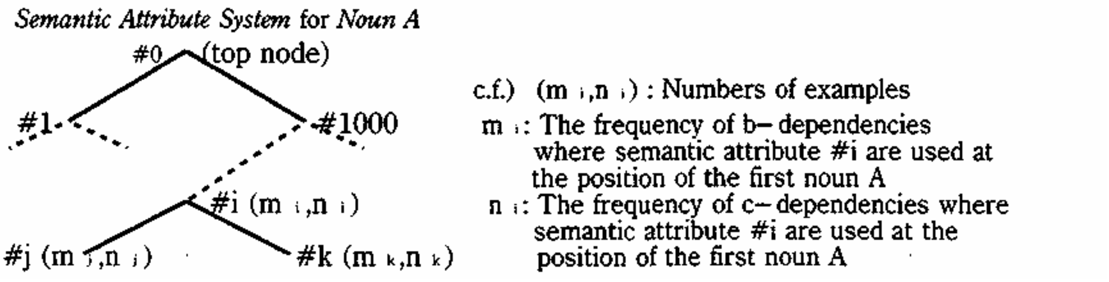
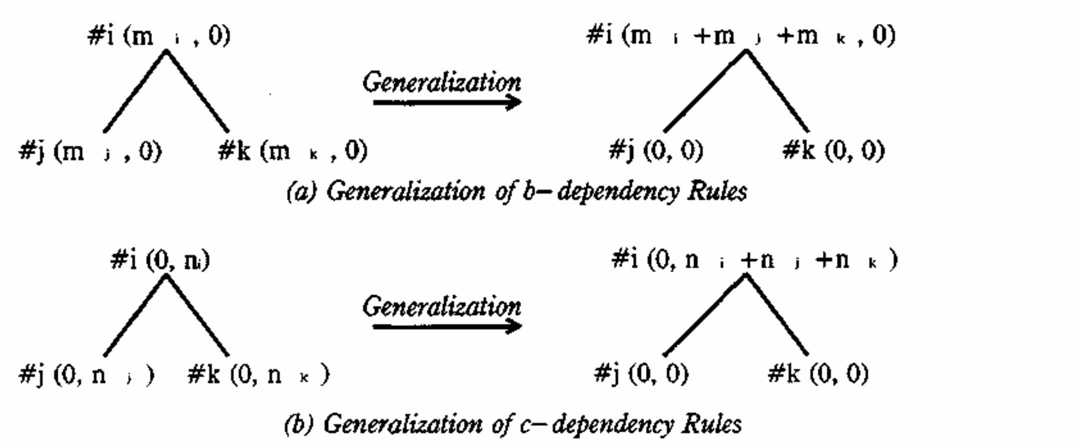

Semantic Attribute System for *Noun A*
#0 (top node)
#1
#1000
#i (m i,n i)
#j (m j,n j)
#k (m k,n k)
c.f.) (m i,n i) : Numbers of examples
m i: The frequency of b– dependencies where semantic attribute #i are used at the position of the first noun A
n i: The frequency of c– dependencies where semantic attribute #i are used at the position of the first noun A

Fig. 3 Extraction of *One— dimensional Rules*

Here, m_i and n_i represent the frequencies of examples for *b-dependency* and *c-dependency* respectively, in which the semantic attribute #i is used. These values can easily be obtained from the learning examples.

If m_i and n_i are calculated from a sufficient number of example data, it is assured that *one-dimensional rules* exist at the nodes where either one of m i and n i is zero. On the contrary, there is no *one-dimensional rule* at the node j where neither of these are zero. For example, the node #i, where $m_i \neq 0$ and $n_i = 0$, yields the dependency rule (i, 0, 0: b). On the other hand, the node #j where m_j =0 and $n_j \neq 0$ yields the dependency rule (j, 0, 0 : c). No rule is generated from the nodes k where $m_k \neq 0$ and $n_k \neq 0$; these nodes are left for extraction by *two-dimensional rules* or *three-dimensional rules.* Here, note that there is no need to generate a rule from the position q where m_q=0 and n_q=0.

As far as not losing accuracy is concerned, the more general and smaller number is the better for dependency rules. Taking notice that an upper node semantically includes lower nodes in *the Semantic Attribute System,* the rules can in many cases be generalized by adding the lower node values to the upper node value.



(a) Generalization of b– dependency Rules

(b) Generalization of c– dependency Rules

Fig. 4 Generalization of *One- dimensional Rules*

Fig. 4 shows the generalization method for the case where node #j and node #k have the same parent node #i. Example *(a)* shows that the three dependency rules, (#i, 0, 0: b), (#j, 0, 0: b) arid (#k, 0, 0: b) can be reduced to one dependency rule, (#i, 0, 0: b). And example *(b)* also shows that the three dependency rules, (#i, 0, 0: c), (#j, 0, 0: c) and (#k, 0, 0: c) can be reduced to dependency rule, (#i, 0, 0: c).

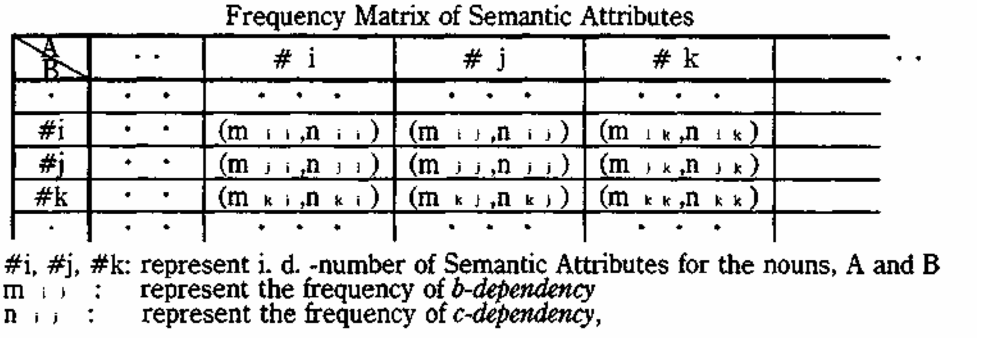
The same goes for Semantic Attribute Systems for noun B and noun C; and dependency rules (0, #i, 0: b) and (0, #i, 0: c) can be obtained for the noun B and (0, 0, #i: b) and (0, 0, #i: c) for the noun C.

(2) *Two-dimensional Rule* Generation

Dependency relationships can be determined by the semantic attributes of two of three nouns. Let us consider a case in which the nouns A and B have semantic constraints. The matrix tables as shown in Fig. 5 are used.

Here, line #i and column #j represent i. d.-numbers of the semantic attributes of noun A and B respectively. And in the matrix, elements (m_{ij}, n_{ij}), m_{ij} and n_{ij} represent the frequency of *b-dependency* and *c-dependency,* respectively.

Similarly to the case of *one-dimensional rules, two-dimensional rules* are generated as shown below. If m_{ij} and n_{ij} are calculated from a sufficient number of example data, it is assured that *two-dimensional rules* can be generated from the matrix elements where one of the values of m_{ij} and n_{ij} is zero. On the contrary, there is no *two-dimensional rule* generated at the elements where neither of these are zero.



Frequency Matrix of Semantic Attributes

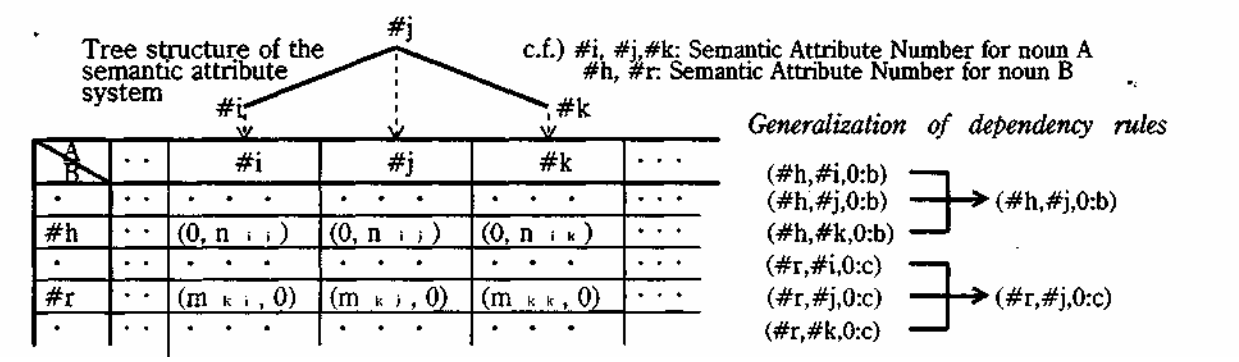
A \ B	· ·	# i	# j	# k		· ·
·	· ·	· · ·	· · ·	· · ·		
#i	· ·	(m_{ii},n_{ii})	(m_{ij},n_{ij})	(m_{ik},n_{ik})		
#j	· ·	(m_{ji},n_{ji})	(m_{jj},n_{jj})	(m_{jk},n_{jk})		
#k	· ·	(m_{ki},n_{ki})	(m_{kj},n_{kj})	(m_{kk},n_{kk})		
·	· ·	· · ·	· · ·	· · ·		

#i, #j, #k: represent i. d. -number of Semantic Attributes for the nouns, A and B
m_{ij} : represent the frequency of *b-dependency*
n_{ij} : represent the frequency of *c-dependency*,

Fig. 5 Extraction of *Two-dimensional Rules*

Thus, if we consider the matrix for the pair of noun A and noun B, dependency rules (#i, *#j,* 0: b) are obtained from the matrix element of #i and #j where m_{ij} =£ 0 and n_{ij} =0. Also, dependency rules (#j, #k, 0: c) are obtained from the matrix element j and k where $m_{jk} = 0$ and $n_{jk} \neq 0$.

The rule generalization can be conducted as shown in Fig. 6 similar to *one-dimensional rules.*



Tree structure of the semantic attribute system
#j
#i
#k
c.f.) #i, #j,#k: Semantic Attribute Number for noun A
#h, #r: Semantic Attribute Number for noun B
Generalization of dependency rules
(#h,#i,0:b)
(#h,#j,0:b)
(#h,#k,0:b)
(#h,#j,0:b)
(#r,#i,0:c)
(#r,#j,0:c)
(#r,#k,0:c)
(#r,#j,0:c)

A \ B	· ·	#i	#j	#k	· · ·
·	· ·	· · ·	· · ·	· · ·	· · ·
#h	· ·	(0, n_{ii})	(0, n_{ij})	(0, n_{ik})	· · ·
·	· ·	· · ·	· · ·	· · ·	· · ·
#r	· ·	(m_{ki}, 0)	(m_{kj}, 0)	(m_{kk}, 0)	· · ·
·	· ·	· · ·	· · ·	· · ·	

Fig. 6 Generalization of *Two- dimensional Rules*

(3) *Three-dimensional Rule* Generation

Three-dimensional rules are obtained from the three-dimensional array in a fashion similar to the case of *two-dimensional rules.* Rules (#i, #j, #k: D) are extracted from the array element (m_{ijk}, n_{ijk}) where either value is zero. Rule generalization is also performed in the same way.

(4) *Exceptional Rule* Generation

In the above three kinds of rule extraction, the noun phrase examples that had been used were deleted from the learning data set. The remainder of the examples conform the exceptional rule set. Each example of the remainder corresponds to an exceptional rule.

(5) Cut off values

Let's consider the reliability of generated rules. The lower dimensional rule covers the wider range so that the lower dimensional rule which are generated from small number of examples will decrease the accuracy of dependency analysis. Then, we use cut off value γ for the above procedure such that no rule is generated from the node which has less than γ examples.

4. Experimental Evaluation

4.1 Procedure of Experiments

First of all, the example data set of 10,000 noun phrases was extracted from the 100 novels (9 million words) in *Sinchou Bunko* using the morphological analysis program ALT-JAWS (NTT 1996). Not only were semantic attributes determined for each of the nouns A, B and C included in the noun phrases listed in *"A Japanese Lexicon",* but correct dependency relationships were also determined for each example.

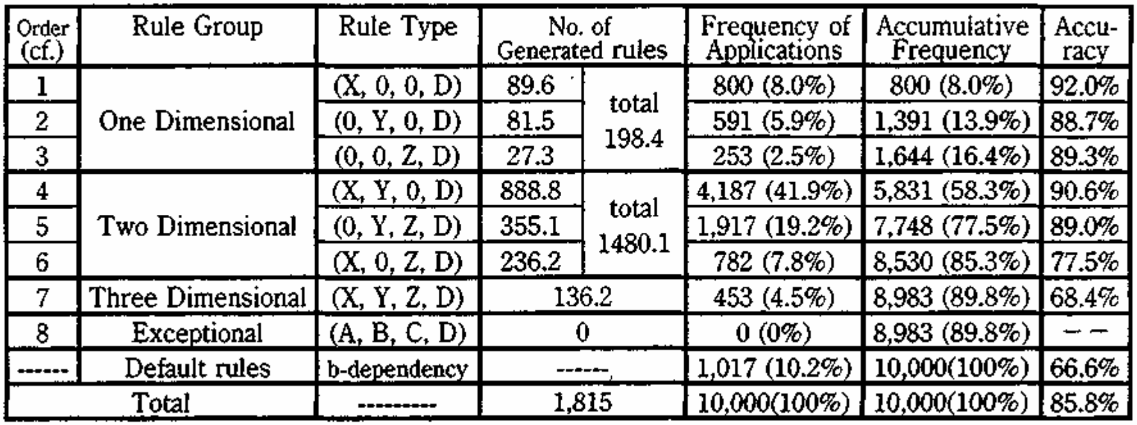
The experiments were conducted by *cross-validation method* as follows. First, the collected examples were separated into 9,000 examples *(the learning set)* for the dependency rule generation and 1,000 examples *(the test set)* for the dependency analysis experiment. Second, according to the proposed method, dependency rules were generated from *the learning set* in the order of *one-, two-, three-dimensional rules* and *exceptional rules.* Notice that every time when *one-, two- and three-dimensional rules* are generated, the examples used for each rule generation are deleted and the examples left behind these three processes are classified as *exceptional rules.* Third, the obtained rules were applied to the dependency analysis of *the test set* in the same order of the rule generations. This procedure was repeated ten times, changing *the learning set* and *the test set.*

In this experiment, *cut off value* are set 10, 2 and 1 for *one-, two-* and *three dimensional rule* generation respectively. And, when there was no generated rule to be applied in the dependency analysis, default rule *(b-dependency* preference) was applied.

4.2 Results and Observations

Table 1 shows the number of dependency rules generated and the results of dependency analysis by using them.

Table 1. Number of Generated Rules and Results of Dependency Analysis by using them



Order (cf.)	Rule Group	Rule Type	No. of Generated rules		Frequency of Applications	Accumulative Frequency	Accu-racy
1	One Dimensional	(X, 0, 0, D)	89.6	total 198.4	800 (8.0%)	800 (8.0%)	92.0%
2		(0, Y, 0, D)	81.5		591 (5.9%)	1,391 (13.9%)	88.7%
3		(0, 0, Z, D)	27.3		253 (2.5%)	1,644 (16.4%)	89.3%
4	Two Dimensional	(X, Y, 0, D)	888.8	total 1480.1	4,187 (41.9%)	5,831 (58.3%)	90.6%
5		(0, Y, Z, D)	355.1		1,917 (19.2%)	7,748 (77.5%)	89.0%
6		(X, 0, Z, D)	236.2		782 (7.8%)	8,530 (85.3%)	77.5%
7	Three Dimensional	(X, Y, Z, D)	136.2		453 (4.5%)	8,983 (89.8%)	68.4%
8	Exceptional	(A, B, C, D)	0		0 (0%)	8,983 (89.8%)	– –
------	Default rules	b-dependency	------		1,017 (10.2%)	10,000(100%)	66.6%
Total		---------	1,815		10,000(100%)	10,000(100%)	85.8%

(cf.) The order of rule generation and their applications for dependency analysis.

(1) Dependency Rule Generation

From Table 1, the results of the rule generations are summarized as follows:

1) Total number of rules within the three dimensional rules which can be defined by using semantic attributes amount to about 1,800.

2) The number of *two-dimensional rules* is high compared to the other rule groups.

3) There are 136 rules which are defined by using all of three semantic attributes.

It is considered that these rules are generated from the noun phrases that *the compositional semantics* does not hold.

4) There was no exceptional rule generated. This means that all of the rules for dependency analysis, in this case, were represented by using *the Semantic Attribute System.*

(2) Dependency Analysis

As for dependency analysis, the following observation can be obtained:

1) Most of the noun phrases (cover ratio is 89.8%) were analyzed by using the generated rules.

2) The accuracy of the *one-dimensional rules* and *the two-dimensional rules* was almost the same. However, the accuracy rate of *three-dimensional rules* was low. This means that the accuracy rates of rules depend on the amount of examples used to generate each of them. The average accuracy rate of these three rule group was 88.0%.

3) No generated rule was applied to 10.2% of the noun phrases then the default rule was applied to them. The accuracy rate of this rule (66.6%) was the lowest. Thus, the final accuracy rate including default rule was 85.8%.

There were many examples included in the data set for which even humans could not uniquely determine the dependency relationships. It is important to notice that the ambiguous portion of this type of noun phrase amounts to almost 10%. Next, majority decision was adopted as the method for determining the correct answer in the experiment. Given these conditions, we can say that the accuracy rate of 85.8% is high and accurate rule sets can be obtained by the proposed method.

5. Concluding Remarks

An automatic dependency rule generation method for noun phrases of the form "A *no* B *no* C" has been proposed. This is one of the most typical of the noun phrases in Japanese that have ambiguities in their dependency relationships. In this method, taking the importance of structural meanings of expressions into consideration, semantic structures for noun phrases were defined using the precise *semantic attribute system,* and four kinds of dependency rules were automatically generated in the order of generality from the examples.

As the results of applying this method to 10,000 noun phrases, 198 rules, 1,480 rules, 136 rules and 0 rules were generated for *one-dimensional, two-dimensional, three-dimensional rules* and *exceptional rules* respectively. Dependency analysis experiments were conducted using these rules, resulting in an accuracy rate of 85.8%.

It is expected that this method can be applied to the dependency rule generation of other types of noun phrases, such as "A *to* B *to* C" (similar to "A and B and C" in English), "A *to* B *no* C" ("C of A and B") and "adjective + A *no/to* B" (adjective + A and/of B).

Acknowledgments

This study was supported by NTT Communication Science Laboratories. The authors would like to thank Mr. S. Shirai for his kind discussion.

References

Almuallim, H., Yamazaki, T., Akiba, Y., Yokoo, A., Kaneda, S. (1994) “A Tool for the Acquisition of Japanese- English Machine Translation Rules Using Inductive Leaning Techniques”, *The 10th Conference of Artificial Intelligence for Application*

Ikehara, S., Miyazaki, M., Shirai, S., Hayashi, Y. (1987): Speaker’s Recognition and Multi-level Machine Translation Method based on It, *Trans. of the IPSJ,* Vol.28, No.12, pp.1269-1279

Ikehara, S., Miyazaki, M., Yokoo, A. (1993): Classification of Linguistic Knowledge for Meaning Analysis in Machine Translations, *Trans. of the IPSJ,* Vol.34, No.l, pp.29-43

Ikehara, S., Miyazaki, M., Shirai, S., Yokoo, A., Nakaiwa, H., Ogura, K., Ooyama, Y., Hayashi, Y.(1997): *Goi-Taikei – a Japanese Lexicon* (Vol. 5), Iwanami, Tokyo

Kobayashi, Tokunaga, Tanaka (1996): Compound Noun Analysis using Semantic Co-occurrences of Nouns, *J. of the Natural Language Processing,* Vol.3, No.l, pp.29-43

Kurohashi, S., Nagao, M. (1992): A Method for Analyzing Conjunctive Structure in Japanese, *Trans. of the IPSJ,* Vol.33, No.8, pp.1022-1031

Matsumoto, Y., Kageyama, T., Nagata, M., Saitou, H., Tokunaga, T.: Words and Dictionary, Iwanami Publisher (1997)

Mizutani, S., Ishiwata, T., Ogino, T., Kaku, N., Kusanagi, H. : Grammar and Meanings, Asakura Publisher (1983)

Nagao, M. (1984): A Framework of a Machine Translation between Japanese and English by Analogy Principle, in A. Elithorn & Banerji, R. (Ed). *Artificial and Intelligence,* North Holland.

NTT (1996): Reference Manual for Morphological Analysis Program ALT-JAWS

Pollard, C., Sag, I. A. : Head-Driven Phrase Structure Grammar, The University of Chicago Press (1994)

Sasaki, M., Sakamoto, H. (1995): Dependency Analysis for Japanese Noun Phrases, *Proc. of the 1st Annual Meeting of the Association for Natural Language Processing,* pp.101-104

Shimazu, A., Naito, S., Nomura, H. (1986): Analysis of Semantic Relations between nouns connected by a Japanese particle "no", *Mathematical Linguistics,* Vol.15, No.7, pp.247-266.

Shimazu, A., Naito, S., Nomura, H. (1987): Semantic Structure Analysis of Japanese Noun Phrases with Adonominal Particles, *Proceedings of the 25th ACL*

Sato, S. (1992): Example based Machine Translation, *Trans. of the IPSJ,* Vol.33, No.6, pp.673-681

Shirai, S., Ikehara, S., Yokoo, A., Kimura J. (1995): Dependency Analysis of Japanese Predicates based on the Hierarchical Semantic Structure, *Trans. of the IPSJ,* Vol.36, No.10, pp.2353-2361

Sumita, E. and Iida, H. (1992): Example-based Transfer of Japanese Adonominal Particles into English, *IEICE Trans. Inf. & Syst.,* pp585-594

Tomiura, Y., Nakamura, T., Hidaka, T. (1995): Semantic Structure of Japanese Noun Phrases "NP no NP", *Trans. of the IPSJ,* Vol.36, No.6, pp.1441-1448.

Utsuro, T., Matsumoto, Y., Nagao, M. (1993): Automatic Acquisition of Case Frames from Parallel Corpus, *Trans. of the IPSJ,* Vol.34, No.5, pp.913-924