A Syntactic Analysis Method of Long Japanese Sentences Based on the Detection of Conjunctive Structures

Sadao Kurohashi*
Kyoto University

Makoto Nagao*
Kyoto University

This paper presents a syntactic analysis method that first detects conjunctive structures in a sentence by checking parallelism of two series of words and then analyzes the dependency structure of the sentence with the help of the information about the conjunctive structures. Analysis of long sentences is one of the most difficult problems in natural language processing. The main reason for this difficulty is the structural ambiguity that is common for conjunctive structures that appear in long sentences. Human beings can recognize conjunctive structures because of a certain, but sometimes subtle, similarity that exists between conjuncts. Therefore, we have developed an algorithm for calculating a similarity measure between two arbitrary series of words from the left and the right of a conjunction and selecting the two most similar series of words that can reasonably be considered as composing a conjunctive structure. This is realized using a dynamic programming technique. A long sentence can be reduced into a shorter form by recognizing conjunctive structures. Consequently, the total dependency structure of a sentence can be obtained by relatively simple head-dependent rules. A serious problem concerning conjunctive structures, besides the ambiguity of their scopes, is the ellipsis of some of their components. Through our dependency analysis process, we can find the ellipses and recover the omitted components. We report the results of analyzing 150 Japanese sentences to illustrate the effectiveness of this method.

1. Introduction

Machine translation systems are gradually being accepted by a wider range of people, and accordingly the improvement of machine translation systems is becoming an urgent requirement by manufacturers. There are many difficult problems that cannot be solved by the current efforts of many researchers. Analysis of long Japanese sentences is one of them. It is difficult to get a proper analysis of a sentence whose length is more than 50 Japanese characters, and almost all the current analysis methods fail for sentences composed of more than 80 characters. By analysis failure we mean the following:

- that no correct analysis is included in the multiple analysis results that are derived from the intrinsic ambiguity of a sentence or by inaccurate grammatical rules;
- that the analysis fails in the middle of the analysis process because an unacceptably large number of parses for a sentence is produced.

* Department of Electrical Engineering, Kyoto University, Kyoto, 606, Japan.

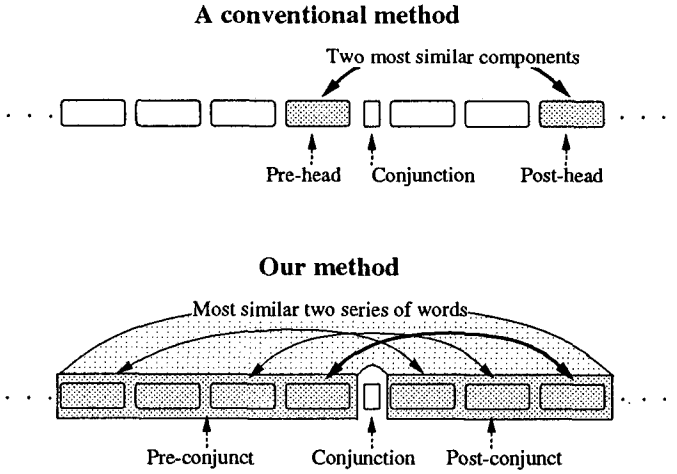


Figure 1
Comparison between a conventional method and our method.

Some researchers have attributed the difficulties to the numerous possibilities of head-dependent relations between phrases in long sentences. But no deeper consideration has ever been given to the reasons for the analysis failure.

A long sentence, particularly in Japanese, very often contains conjunctive structures. These may be either conjunctive noun phrases or conjunctive predicative clauses. Among the latter, those made by the *renyoh forms* of predicates (the ending forms that mean connection to another right predicate) are called *renyoh chuushi-ho* (see example sentence (iv) of Table 1). A renyoh chuushi-ho appears in an embedded sentence to modify nouns and is also used to connect two or more sentences. This form is used frequently in Japanese and is a major cause of structural ambiguity. Many major sentential components are omitted in the posterior part of renyoh chuushi expressions, thus complicating the analysis. For the successful analysis of long sentences, these conjunctive phrases and clauses, including renyoh chuushi-ho, must be recognized correctly. Nevertheless, most work in this area (e.g., Dahl and McCord 1983; Fong and Berwick 1985; Hirschman 1986; Kaplan and Maxwell 1988; Sag et al. 1985; Sedogbo 1985; Steedman 1990; Woods 1973) has concerned the problem of creating candidate conjunctive structures or explaining correct conjunctive structures, and not the method for selecting correct structures among many candidates. A method proposed by some researchers (Agarwal and Boggess 1992; Nagao et al. 1983) for selecting the correct structure is, in outline, that the two most similar components to the left side and to the right side of a conjunction are detected as two *conjoined heads* in a conjunctive structure. For example, in "John enjoyed the book and liked the play" we call the verbs "enjoyed" and "liked" conjoined heads; "enjoyed" is the *pre-head*, and "liked" the *post-head*. We also call "enjoyed the book" *pre-conjunct*, and "liked the play" *post-conjunct*. In Japanese, the word preceding a conjunction is the pre-head, and the post-head that is most similar to the pre-head is searched for (Nagao et al. 1983) (see the upper part of Figure 1). In English, conversely, the phrase following the conjunction is the post-head, and the pre-head is searched for in the same way (Agarwal and Boggess 1992).

However, two conjoined heads are sometimes far apart in a long sentence, making this simple method clearly inadequate.

Human beings can recognize conjunctive structures because of a certain, but sometimes subtle, similarity that exists between conjuncts. Not only the conjoined heads, but also other components in conjuncts, have some similarity, and furthermore, the pre- and post-conjuncts have a structural parallelism. A computational method needs to recognize this subtle similarity in order to detect the correct conjunctive structures. In this investigation, we have developed an algorithm for calculating a similarity measure between two arbitrary series of words from the left and the right of a conjunction and selecting the two most similar series of words that can reasonably be considered as composing a conjunctive structure (see the lower part of Figure 1). This procedure is realized using a dynamic programming technique.

In our syntactic analysis method, the first step is the detection of conjunctive structures by the above-mentioned algorithm. Since two or more conjunctive structures sometimes exist in a sentence with very complex interrelations, the second step is to adjust tangled relations that may exist between two or more conjunctive structures in the sentence. In this step conjunctive structures with incorrect overlapping relations, if they exist, are found and retrials of detecting their scopes are done. The third step of our syntactic analysis is a very common operation. Japanese sentences can best be explained by *kakari-uke*, which is essentially a dependency structure. Therefore our third step, after identifying all the conjunctive structures, is to perform dependency analyses for each phrase/clause of the conjunctive structures and the dependency analysis for the whole sentence after all the conjunctive structures have been reduced into single nodes. The dependency analysis of Japanese is rather simple. A component depends on a component to its right (not necessarily the adjacent component), and the suffix (postposition) of a component indicates what kind of element it can depend on. More than one head-dependent relation may exist between components, but by introducing some heuristics, we can easily get a unique dependency analysis result that is correct for a high percentage of cases. A serious problem regarding conjunctive structures, in addition to the ambiguity of their scopes, is the ellipses in some of their components. Through the dependency analysis process outlined, we are able to find the ellipses occurring in the conjunctive structures and supplement them with the omitted components.

2. Types of Conjunctive Structures and Their Ambiguities

In Japanese, *bunsetsu* is the smallest meaningful sequence consisting of an independent word (IW; nouns, verbs, adjectives, etc.) and accompanying words (AW; copulas, postpositions, auxiliary verbs, and so on). A bunsetsu whose IW is a verb or an adjective, or whose AW is a copula, functions as a predicate and thus is called a predicative bunsetsu (PB). A bunsetsu whose IW is a noun is called a nominal bunsetsu (NB).

Conjunctive structures (CSs) that appear in Japanese are classified into three types (Shudo et al. 1986). The first type is the *conjunctive noun phrase*. We can find these phrases by the words listed in Table 1-a. Each conjunctive noun can have adjectival modifiers (Table 1-ii) or clausal modifiers (Table 1-iii).

The second type is the *conjunctive predicative clause*, in which two or more predicates in a sentence form a coordination. We can find these clauses by the renyoh forms of predicates (Table 1-iv) or by the predicates accompanying one of the words in Table 1-b (Table 1-v).

The third type is a CS consisting of parts of conjunctive predicative clauses. We call this type an *incomplete conjunctive structure*. We can find these structures by the

Table 1
Types of conjunctive structures

Conjunctive noun phrases	
Words indicating conjunctive noun phrases:	
(a)	,[comma]* TO MO YA TOKA KATSU OYOBI NARABINI (*and*) KA ARUIWA MATAWA MOSHIKUWA (*or*) DAKEDE⟨WA⟩NAKU(*not only .. but also ..*)
Example:	
(i)	... KAISEKI(*analysis*) TO(*and*) SEISEI(*generation*) WO ...
(ii)	... GEN-GENGO(*source language text*) NO(*of*) KAISEKI(*analysis*) TO(*and*) AITE-GENGO(*target language text*) NO(*of*) SEISEI(*generation*) WO ...
(iii)	... GEN-GENGO(*source language text*) WO KAISEKI-SURU (*analyzing*) SHORI (*processing*)TO(*and*) AITE-GENGO(*target language text*) WO SEISEI-SURU(*generating*) SHORI(*processing*) WO ...
Conjunctive predicative clauses	
Words indicating conjunctive predicative clauses:	
(b)	TOKA SHI OYOBI NARABINI (*and*) KA ARUIWA MATAWA MOSHIKUWA (*or*) GA NONI-TAISHI⟨TE⟩ KEREDOMO (*but*) DAKEDE⟨WA⟩NAKU(*not only .. but also ..*) ZU-NI(*without ..ing*)
Example:	
(iv)	... GEN-GENGO(*source language text*) WO KAISEKI-SHI(*analyzing*), AITE-GENGO(*target language text*) WO SEISEI-SURU(*generating*) (SHORI(*processing*) WO ...).
(v)	... KAISEKI(*analysis*) DE-WA(*for*) RIYOU-SURU(*use*) GA(*but*), SEISEI(*generation*) DE-WA(*for*) RIYOU-SHI-NAI(*do not use*) (TO-IU(*as*) ...).
Incomplete conjunctive structures	
Words indicating incomplete conjunctive structures:	
(c)	,[comma]* OYOBI NARABINI (*and*) ARUIWA MATAWA MOSHIKUWA (*or*)
Example:	
(vi)	... ZENSHA(*the former*) WO KAISEKI(*analysis*) NI(*for*), KOUSHA(*the latter*) WO SEI-SEI(*generation*) NI(*for*) ...

Characters in '⟨⟩' are optional. Japanese postposition "WO" marks the object case.
*A noun directly followed by a comma indicates a conjunctive noun phrase or an incomplete conjunctive structure.

correspondence of case-marking postpositions (Table 1-vi: ".. WO .. NI, .. WO .. NI"). However, sometimes the last bunsetsu of the pre-conjunct has no case-marking postposition (e.g., "NI" can be omitted in the bunsetsu "KAISEKI-NI" in Table 1-vi), just followed by one of the words listed in Table 1-c. In such cases we cannot distinguish this type of CS from conjunctive noun phrases by seeing the last bunsetsu of the pre-conjunct. However, this does not matter, as our method handles the three types of CSs in almost the same way in the stage of detecting their scopes, and it exactly distinguishes incomplete conjunctive structures in the stage of dependency analysis.

For all of these types, it is relatively easy to detect the presence of a CS by looking for a *distinctive key bunsetsu* (we call this a KB) that accompanies a word indicating a CS listed in Table 1 or has the renyoh forms (the underlined bunsetsus are KBs in

Table 1). A KB lies last in the pre-conjunct and is a pre-head. However, it is difficult to determine which bunsetsu sequences on both sides of the KB constitute pre- and post-conjuncts. That is, it is not easy to determine which bunsetsu to the left of a KB is the leftmost bunsetsu of the pre-conjunct (we call this starting bunsetsu SB) and which bunsetsu to the right of a KB is the rightmost bunsetsu of the post-conjunct (this ending bunsetsu is called EB and is a post-head). The bunsetsus between these two extreme bunsetsus constitute the *scope* of the CS. In detecting a CS, it is most important to find the post-head (that is, the EB) among many candidates in a sentence; e.g., in a conjunctive noun phrase, all NBs after a KB are candidates (we call such a candidate bunsetsu a CB). However, our method searches not only for the most plausible EB, but also for the most plausible scope of the CS.

3. Detection of Conjunctive Structures

We detect the scope of CSs by using a wide range of information before and after a KB. An input sentence is first divided into bunsetsus by conventional morphological analysis. Then we calculate similarities in all pairs of bunsetsus in the sentence. After that, we calculate the similarities between two series of bunsetsus on the left and right of the KB by combining the similarity scores for pairs of bunsetsus. Then, as a final result, we choose the two most similar series of bunsetsus that can reasonably be considered as composing a CS. We will explain this process in detail in the following sections.

In detecting CSs, it is necessary to take many factors into consideration, and it is important to give the proper weight to each factor. The scoring system described hereafter was first hypothesized and then manually adjusted through experiments on 30 training sentences containing CSs. These parameters would not be the best, and statistical investigations of large corpora would be preferable. However, these parameters are good enough to get reasonably good analysis results, as shown in the experiments section, and to show the appropriateness of our method.

3.1 Similarities between Bunsetsus

First, we calculate similarities for all pairs of bunsetsus in the sentence. An appropriate similarity value between two bunsetsus is given by the following process:

1. If the parts of speech of IWs are equal, give 2 points as the similarity value, and go to step 2. When the parts of speech of IWs are not equal and both bunsetsus are PBs, give 2 points, but do not add other points (i.e., end the scoring process).
2. If IWs match (by character level) each other exactly, add 10 points and go to step 5. If IWs are conjugated, infinitives are compared.
3. If both IWs are nouns and they match partially at the character level, add the number of matching characters $\times$ 2 points.
4. Add points for semantic similarities by using the thesaurus *Bunrui Goi Hyou* (BGH; National Language Research Institute 1964). The BGH has a six layer abstraction hierarchy, and more than 60,000 words are assigned to the leaves of it. If the most specific common layer between two IWs is the kth layer and if k is greater than 2, add $(k - 2) \times 2$ points. If either or both IWs are not contained in the BGH, no addition is made. Matching of the generic two layers is ignored to prevent too vague matching in a

broader sense. The maximum sum of similarity values that can be added by step 3 and this step is 10 points.

5. If some of the AWs match, add the number of matching AWs × 3 points.

For example, the similarity value between "TEISEI(*revision*)SHI(*do*)," and "KEN-SHUTSU(*detection*)SURU(*do*)" is calculated as

2(match of parts of speech) + 2(match by BGH) + 3(match of one AW) = 7 points.

The similarity value between "TEI-SUIJUN-GENGO(*low level language*)," and "KOU-SUIJUN-GENGO(*high level language*)TO(*and*)" is

2(match of parts of speech) + 8(match of four kanji characters: "SUIJUN-GENGO") = 10 points.

Since the BGH does not contain technical terms, similarity points cannot be given to them by the BGH. However, technical terms are often compound words, and those having similar meanings often contain the same words. For such technical terms, some similarity points can be given according to the degree of partial character matching by step 3, as for the latter example.

3.2 Similarities between Two Series of Bunsetsus

Our method detects the scope of a CS by finding the two series of bunsetsus from before and after the KB that have the greatest similarity. To measure the similarity score between two series of bunsetsus, we have developed a method using a triangular matrix, A, as shown in Figure 2 (Figure 7 and Figure 8 are concrete examples):

$$A = (a(i,j))\ (0 \leq i \leq l;\ i \leq j \leq l),$$

where l is the number of bunsetsus in a sentence. Here, each diagonal element $a(i,i)$ is the ith bunsetsu in a sentence (hereafter denoted B_i) and every other element $a(i,j)$ $(i < j)$ is the similarity value between bunsetsu B_i and bunsetsu B_j calculated by the process just described.

In detecting a CS whose KB is the nth bunsetsu (B_n), we consider only a *partial matrix* (denoted A_n) that is the upper right part of B_n (Figure 2):

$$A_n = (a(i,j))\ (0 \leq i \leq n;\ n+1 \leq j \leq l).$$

For specifying candidate pre- and post-conjuncts and measuring their similarity, we define a *path* in A_n (Figure 2):

$$\text{path} ::= (a(p_1,m), a(p_2,m-1), \ldots, a(p_{m-n}, n+1)),$$
$$\text{where } n+1 \leq m \leq l,\ a(p_1,m) \neq 0,\ p_1 = n,\ p_i \geq p_{i+1} (1 \leq i \leq m-n-1).$$

That is, a path is a series of elements from a non-zero element in the lowest row in A_n to the element in the leftmost column in A_n. It has only one element in each column and extends toward the upper left. The series of bunsetsus on the left side of the path (sb_1 in Figure 2) and the series under the path (sb_2 in Figure 2) are candidate conjuncts for a KB, B_n. When a KB is an NB, NBs after it are CBs; when a KB is a PB, PBs after it are CBs. To satisfy this condition, a path starts from a non-zero element that shows the

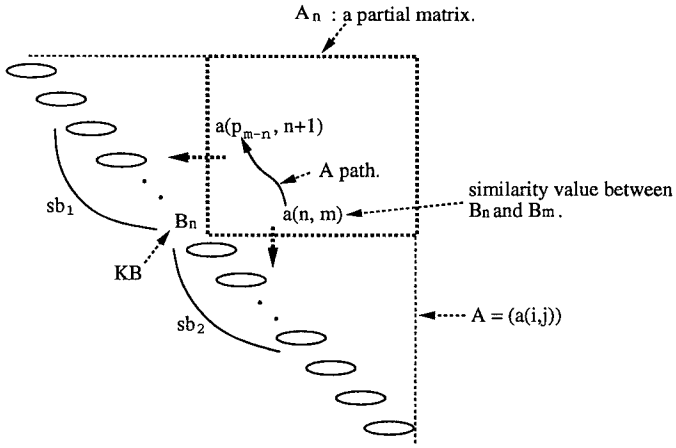


Figure 2
A method using a triangular matrix.

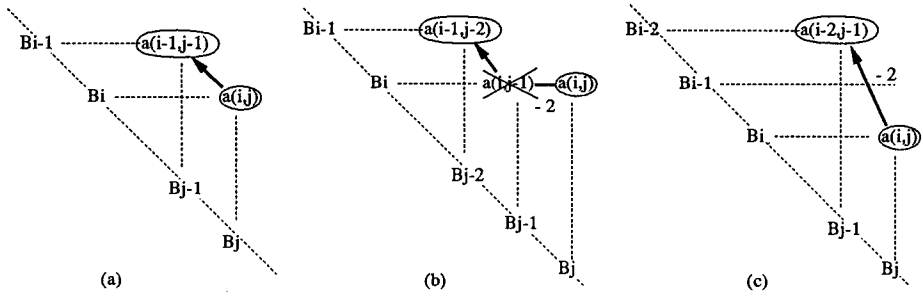


Figure 3
The way of calculating path scores.

correspondence of a KB to a CB (note the first process giving the similarity between two bunsetsus).

We calculate a *path score*, which shows the similarity between two candidate conjuncts specified by the path, using the following five criteria:

1. Basically, the score of a path is the sum of each element's points on the path. For example, the similarity score between phrase B_{i-1} B_i and phrase B_{j-1} B_j in Figure 3a can be expressed by $a(i-1,j-1)+a(i,j)$.
2. When candidate conjuncts have one-to-one bunsetsu correspondences, the path extends by jumping over rows one by one, and its score can be obtained simply by the previously described method. However, when an extra bunsetsu is inserted in one conjunct, like " ... BOKU-NO(*my*) AKAI(*red*) PEN(*pen*) TO(*and*) KARE-NO(*his*) ENPITSU(*pencil*) ...," the path extends toward the left adjacent element (i.e., horizontally) or extends by jumping over two or more rows. To handle a bunsetsu

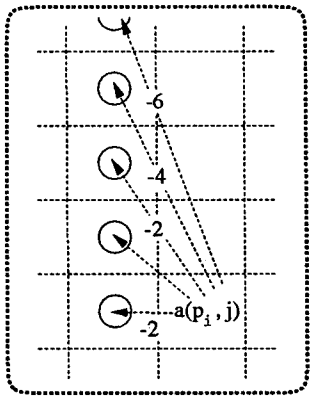


Figure 4
Penalty points.

insertion in pre-conjunct and that in post-conjunct symmetrically, when a part of the path is horizontal $(a(i,j), a(i,j-1))$, the element's points $a(i,j-1)$ are not added to the path score (Figures 3b and 3c).[1]

3. Since a pair of conjunctive phrases/clauses often exhibit structural similarity, we hypothesize that analyses of CSs which maximize corresponding bunsetsus tend to lead to a correct resolution of the conjunctive scope. By this hypothesis, we impose penalty points on the pairs of elements in the path that cause one to multiple bunsetsu correspondence, giving priority to CSs that are constructed of components of the same size. Penalty points for $(a(p_i,j), a(p_{i+1}, j-1))$ calculated by the following formula are subtracted from the path score (Figure 4):

$$|p_i - p_{i+1} - 1| \times 2.$$

Note that these penalty points are also symmetrical, as shown in Figures 3b and 3c.

4. Since each phrase in the CS has a certain coherence of meaning, special words that separate different meanings in a sentence often limit the scope of a CS. If candidate conjuncts specified by a path include such words, we impose penalty points on the path so that the possibility of selecting such a path is reduced. We define five *separating levels* (SLs) for bunsetsus, which express the strength of separating meanings in a sentence (Table 2; see Table 1), by observing sentences containing CSs. If candidate conjuncts contain a bunsetsu whose SL is equal to the KB's SL or higher, we reduce the path score by

$$(\text{SL of the bunsetsu} - \text{KB's SL} + 1) \times 7.$$

1 This is caused by the path definition that a path has one element in each column. However, this definition fits the dynamic programming method described later, which calculates scores for partial paths column by column.

Table 2
Separating levels (SLs)

Level	Conditions for bunsetsu
5	Being the KB of a conjunctive predicative clause, or accompanying the topic-marking postposition "WA" and a comma.
4	Accompanying a case-marking postposition (e.g., "GA," "WO") and a comma, or being an adverb accompanying a comma.
3	Being the renyoh form of a predicate not accompanying a comma, or accompanying the topic-marking postposition "WA."
2	Being the KB of a conjunctive noun phrase accompanying a comma.
1	Accompanying a comma, or being the KB of a conjunctive noun phrase not accompanying a comma.

Table 3
Words for bonuses

	Conjunctive noun phrases
Last AW	NADO(*and so on*)
Next IW	KAKU(*each*) SHURUI(*sort*) ···TSU(numeral) KUMI(*set*) TSUI(*pair*) RYOUHOU(*both*)
	Conjunctive predicative clauses
Last AW	TAME-NI(*in order to*) TAME-NO(*in order to*) TO-IU(*as*) TO-ITTA(*as*) YOUDA(*like*) NADO(*and so on*)
Next IW	KOTO(*that*) MONO(*that*) TOKI(*when*) HOU-HOU(*way*) HOU-SHIKI(*way*) SHU-HOU(*way*)

However, two high SL bunsetsus corresponding to each other often exist in a CS, and these do not limit the scope of the CS, like "X TO-SHITE WA(*As to* X), ... DE-ARI(*be*), Y TO-SHITE WA(*as to* Y), ... DE-ARU(*be*)." To take this into consideration, penalty points for corresponding high SL bunsetsus are not given to paths. For high SL bunsetsus, B_i and B_j, to be corresponding, they have to be of the *same type*, and the path contains the element $a(i,j)$. We define two bunsetsus to be of the same type if:

- their IWs are of the same part of speech;
- when they are conjugated, they have the identical form;
- when they contain AWs, their AWs are identical.

For example, "KARE(*he*)-WA" and "KANOJO(*she*)-WA" are of the same type (noun + postposition "WA"). So are "HASHIREBA(*if run*)" and "ARUKEBA(*if walk*)" (conditional form of verb). These penalty points can be imposed on pairs of elements in a path, namely, extension steps of a path separately because each extension step of a path takes some bunsetsus in candidate conjuncts.

5. Some words frequently are the AW of the last bunsetsu in a CS or the IW following it. These words are shown in Table 3. Bonus points (6 points) are given to paths that have the CS ending with one of the words in Table 3.

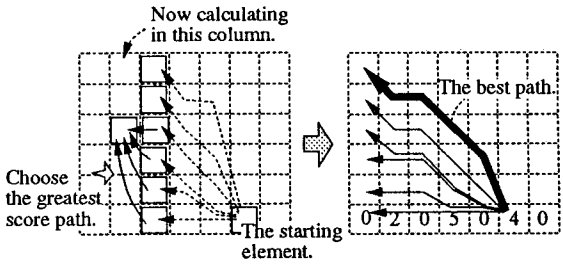


Figure 5
The best path from an element.

3.3 Finding the Conjunctive Structure Scope

As described in the preceding subsection, a path score is composed of points for its elements, penalty points for every path extension, and bonus points for its starting position. The key aspect is that these points can be calculated for every extension step of a path independently. For this reason, the greatest score path can be searched for by using dynamic programming method.

Calculation is performed column by column going left from a non-zero element in the lowest row in A_n to the leftmost column in A_n. For each element in a column, the best partial path reaching it is found by extending the partial paths from the previous column and choosing the greatest score path (the left part of Figure 5). In extending partial paths, elements' points and penalty points are given to paths step by step. Then, among the paths to the leftmost column, the path that has the greatest score becomes the best path from the starting non-zero element (the right part of Figure 5). Of all the best paths from all the non-zero lowest row elements, the path that has the greatest path score (the maximum path) is chosen as defining the scope of the CS; i.e., the series of bunsetsus on the left side of the maximum path (pre-conjunct) and the series of bunsetsus under it (post-conjunct) are conjunctive (Figure 6).

An EB (the last bunsetsu in the post-conjunct) corresponds to a KB (the last bunsetsu in the pre-conjunct), and it follows from the definition of a path that the EB has a certain similarity to the KB. On the other hand, when there are modifiers in both conjuncts, an SB shows where the leftmost modifier starts in its pre-conjunct. Since the modifiers in the pre-conjunct and those in the post-conjunct usually do not correspond exactly, an SB is determined mainly on the basis of the balance between pre- and post-conjuncts and is not always detected precisely. This problem is managed in the next stages when the relations between CSs in a sentence are adjusted and when a dependency structure is constructed (described in Section 4.1 and Section 5.2).

3.4 Examples of Detection of Conjunctive Structures

Two examples of detecting CSs are shown in Figures 7 and 8. A chain of matrix elements with the same letters shows the maximum path for the KB marked with this letter and '>.'

In the example sentence in Figure 7, the conjunctive predicative clause is detected correctly owing to the penalty points for the SL of the topic-marking postposition "WA" and the comma in the bunsetsu "KAISHOU-SURU-TAME-NI-WA(*in order to solve*)," which is outside of the CS, and owing to the bonus points for the IW "KOTO(*that*)" in the next right bunsetsu of the CS.

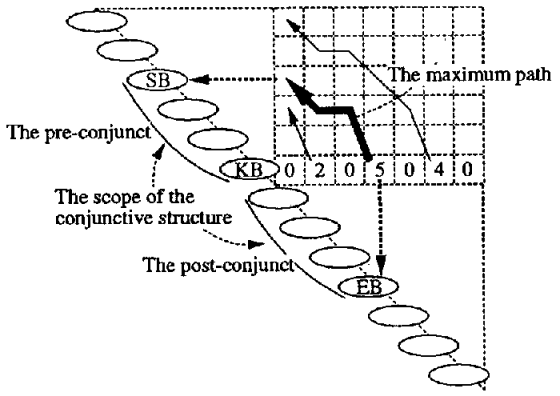


Figure 6
The maximum path specifying a conjunctive structure.

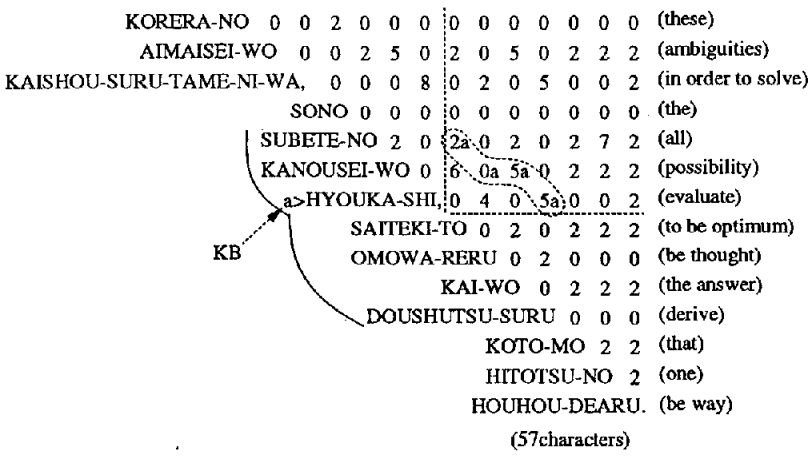


Figure 7
An example of detecting conjunctive structures.

In the sentence illustrated in Figure 8, the conjunctive noun phrase, in which three nouns are conjoined, is detected correctly (chains of 'a' and 'b'). Consecutive overlapping CSs express a CS consisting of more than two conjuncts and will thus be merged into one CS (as described in Section 4). In this example, the conjunctive predicative clause that contains the conjunctive noun phrase is also detected correctly (the 'c' chain).

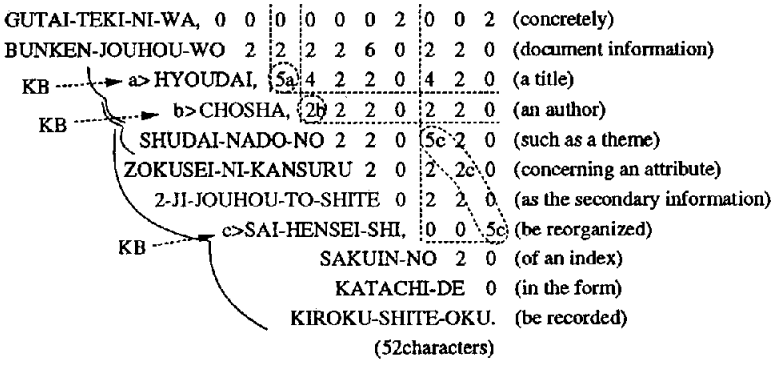


Figure 8
An example of detecting conjunctive structures.

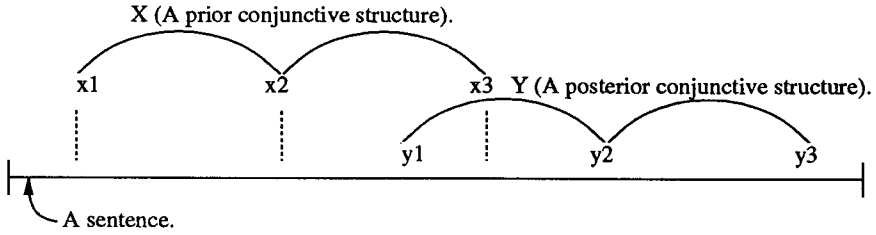
4. Reduction of a Long Sentence with Conjunctive Structures

In a long Japanese sentence two or more CSs often exist, overlapping with each other. In such cases, we have to adjust their relations in a sentence after their scopes have been detected. This adjustment is done by checking relations in all pairs of CSs and merging all the relations. Through this adjustment process, CSs consisting of three or more conjuncts are detected. Furthermore, CSs with incorrect relations, if they exist, are found, and retrials of detecting their scopes are done. As a result of this adjustment process, we get a reduced sentence form. The details of these processes will be given in the following section.

4.1 Relations between Two Conjunctive Structures

The scope of a CS is represented by a three-tuple: {position of SB, position of KB, position of EB}. Let us suppose that two CSs exist in a sentence; the prior one, X, has a scope represented by {x1, x2, x3}, and the posterior one, Y, has a scope represented by {y1, y2, y3} (see Figure 9). When two CSs are detected by the previously described dynamic programming method as overlapping each other, in this case $y1 \leq x3$, there is a variety of possible cases according to the relation among x1, x2, and y1 and that among y2, y3, and x3 as shown in Figure 9. These 16 possible cases of two CSs overlapping each other are classified into three different relations, and the correction of CSs is performed for each relation in the following way:

> **Brother relation** (case F in Figure 9): In the previous step of detecting the scopes of CSs, a CS that consists of more than two conjuncts is detected as composed of consecutive CSs, each of which consists of two conjuncts. In this case, two adjoining CSs have a brother relation. Consecutive CSs that are in a brother relation are merged into one CS.
>
> **Parent–child relation** (cases A, B, C, D, E, G, H, M, and N in Figure 9): Another actual relation between two CSs is a parent–child relation, in which a pre- or post-conjunct of a CS includes another CS. Cases D, H,



y1 \ x3	x3<y2	x3=y2	y2<x3<y3	y3≦x3
x2+1<y1	A ○	E ○	I ×	M ○
y1=x2+1	B ○	F ○	J ×	N ○
x1<y1≦x2	C ○	G ○	K ×	O ×
y1≦x1	D ○	H ○	L ×	P ×

Figure 9
A relation between two conjunctive structures.

M, and N illustrate this relation. Cases A, B, C, and G fall into this category when a pre-conjunct of a posterior CS (Y) is extended to the left to include a prior CS (X), because X is considered to be a modifier in Y's pre-conjunct.[2] Case E also falls into the parent–child category by extending X's post-conjunct to the right to include Y. This is because the EB of the extended X (that is, Y's EB) can correspond to X's KB through the EB of the original X (that is, Y's KB). Apart from case E, a post-conjunct is not extended to the right. Therefore, cases I and J do not come into this relation.

Incorrect relation (cases I, J, K, L, O, and P in Figure 9): These relations do not exist in actual sentences and are caused only by incorrect detection of CSs. Therefore, a retrial of detecting their scopes is done in the way described in the following section.

2 Note that an SB is not always detected precisely by the previously described dynamic programming method, whereas an EB (corresponding to a KB) is.

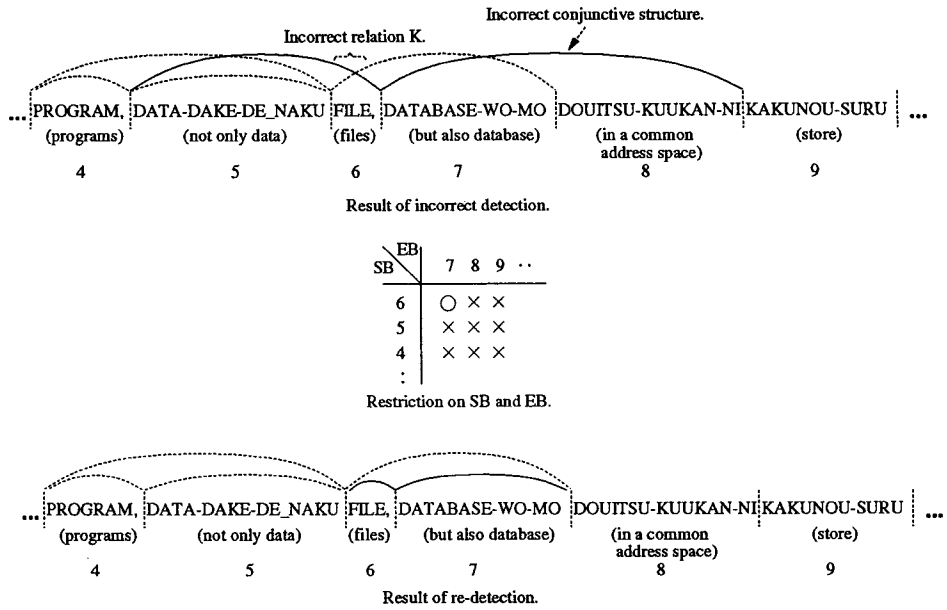


Figure 10
An example of redetecting a conjunctive structure under an incorrect relation between conjunctive structures.

4.2 Retrial of Detecting Overlapping Conjunctive Structures with Incorrect Interrelations

If two CSs have a incorrect relation in a sentence as described earlier, a retrial is conducted to get the proper CSs. In this case, the CS that has the larger CS score is regarded as correct, and a new CS concerning the KB whose old CS had a smaller score is detected so that the new CS does not have an incorrect relation with the other CS. For example, when the CSs X and Y (X precedes Y) have an incorrect relation and Y has a smaller score than X, the range of y1 and y3, which will give a correct relation with x1, x2, x3, and y2, is determined from the relations in Figure 9.[3] Then, the new CS is identified whose similarity score is the greatest in this restricted range of y1 and y3, ignoring paths that start from and end with elements outside the restricted range in the dynamic programming method. An example of redetecting a CS is shown in Figure 10. In this example only a pair of y1 and y3, (6, 7), gives a correct relation so that the scope of the CS is determined uniquely without the dynamic programming method.

If there exist two or more CS pairs in a sentence that all have incorrect relations, the redetection is done on the pair whose difference of scores is the greatest. Whenever the redetection is done on one pair of CSs, the relations of all pairs of CSs in a sentence are checked and contradictory relations are corrected by the previously described process. This continues till no pair of CSs with an incorrect relation exists in the sentence.

3 Calculation is done for each correct case in the matrix in Figure 9. For example, as for case A, the range of y1 and y3 is determined, satisfying $x2 + 1 < y1$ and $x3 < y2$ for given x1, x2, x3, and y2.

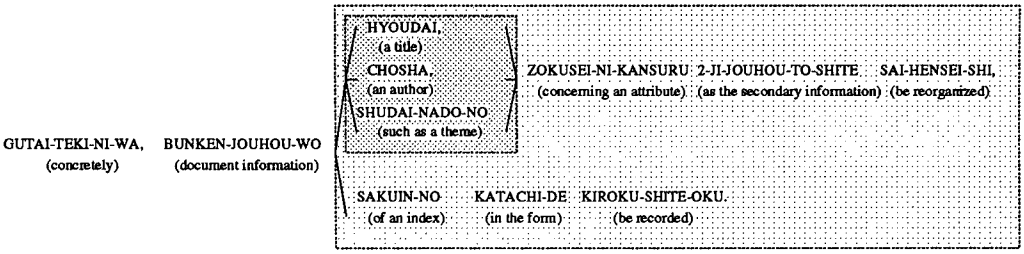


Figure 11
An example of a reduced sentence.

4.3 An Example of Reduction of a Sentence

As for the sentence in Figure 8, the following CSs are detected:

- CS1: [HYOUDAI(*a title*),]-[CHOSHA(*an author*),] ('a>' in Figure 8)
- CS2: [CHOSHA(*an author*),]-[SHUDAI-NADO-NO(*such as a theme*)] ('b>' in Figure 8)
- CS3: [SHUDAI-NADO-NO(*such as a theme*) · · ·SAI-HENSEI-SHI(*be reorganized*),]-
  [SAKUIN-NO(*of an index*)· · ·KIROKU-SHITE-OKU(*be recorded*).] ('c>' in Figure 8)

Because CS1 and CS2 are found to be in a brother relation by checking their overlap relation, they are merged into one CS (CS1-2: [HYOUDAI(*a title*),]-[CHOSHA(*an author*),]- [SHUDAI-NADO-NO(*such as a theme*)]). Then, because CS3 is found to be a parent CS of CS2, that is, a parent CS of CS1–2, its pre-conjunct is extended to contain CS1–2. As a result of this process, the reduced form of a sentence is obtained as shown in Figure 11.

5. Dependency Analysis of a Sentence and Supplementing for Ellipses

As described in the preceding sections, information about CSs can be used to reduce a sentence into a simpler form. Consequently, a dependency structure of an entire sentence can be obtained by applying relatively simple head-dependent rules to CSs and the sentence. Another serious problem regarding CSs, in addition to the ambiguity of scope, is the ellipses that may occur in the components of CSs. We recover the omitted components in the stage of dependency analysis. We will explain this process in the following.

5.1 Dependency Analysis

In this paper, the goal of the syntactic analysis is to transform a sentence into a dependency tree structure in which a dependent bunsetsu node is placed as a child node of its head bunsetsu node. In a Japanese sentence, because each bunsetsu depends on one of the bunsetsus to the right of it, a sentence can be transformed into a tree whose root node is the last bunsetsu in the sentence. This left-to-right head-dependent relation is characteristic of the sentential structure of Japanese, and the dependency analysis fits this very well.

First, each conjunct of the CSs is analyzed. If there are two or more CSs in a nested structure in a sentence (i.e., having parent–child relations), each CS is analyzed from the innermost CS in the order of nesting level. Then finally, the main sentential component is analyzed. Because the pre- and post-conjuncts have their own consistent structures and meanings, they are parsed independently into dependency trees. The root nodes of these trees are the KB and the EB (the last bunsetsu of each conjunct).[4] After analyzing a CS, a new node, called the *CS node*, is created that has two child nodes, KB and EB. The CS node inherits the property of the EB when it depends on a bunsetsu to the right of it, and it inherits the property of the KB and the EB when it governs a bunsetsu to the left of it. In the *next level analysis* (the term we give to the analysis of its parent CS or of the whole sentence if no parent CS exists), the CS node is handled as a symbol. This means that bunsetsus outside a CS can no longer depend on bunsetsus in it, except the KB and the EB. Even in the case of a CS that consists of more than two conjuncts, the same analysis takes place, except that the dependency tree of the CS is composed of more than two sub-trees into which each conjunct is parsed.

Parsing a series of bunsetsus in a certain range (conjuncts of CSs, or a whole sentence after merging all the CSs into CS nodes) is performed in the following way. The head bunsetsu is determined from right to left for each bunsetsu in the range of bunsetsus to the right of it with a no-cross condition.[5] The type of bunsetsu as a head is classified into two types, NB and PB.[6] Whether a bunsetsu depends on NB or PB is determined by the conjugation of its IW or by the type of its AW. For example, an NB with a postposition "NO" can depend on an NB, and a conditional form of a PB (ending with "BA") can depend on a PB. When a bunsetsu can depend on two or more bunsetsus in the range, its head is determined by the following heuristics:

- In most cases a bunsetsu depends on its nearest head in Japanese. Therefore, a bunsetsu is regarded as depending on its *nearest* head except in following two cases.
- Because the postposition "WA" marks the topic of a sentence, a bunsetsu accompanying it usually depends on the last predicate that is the main predicate in a sentence. Thus, such a bunsetsu is regarded as depending on the *last* head in the analysis range.
- A comma in a sentence shows a separation of meaning, and the bunsetsu accompanying a comma usually depends on a bunsetsu farther away than the nearest one. Based on our observation we consider such a bunsetsu to depend on the *second nearest* head.

These rules are rather simple, but they are still useful when applied to the reduced form of a sentence, as shown in the discussion of the experiments.

We illustrate this process for the sentence in Figure 12. At first, the CS [HYOUDAI(*a title*),]-[CHOSHA(*an author*),]- [SHUDAI-NADO-NO(*such as a theme*)] is analyzed; because each conjunct consists of only one bunsetsu, the analysis results only in creating

4 In the case of incomplete conjunctive structures, such as in Table 1-vi, neither conjunct can be parsed into a dependency tree, as it contains no predicate that should become the root node of a dependency tree. A way of dealing with this problem is described in Section 5.3.

5 In Japanese, head-dependent relations do not cross each other, that is, when B_i depends on B_j, B_k $(k < i)$ cannot depend on bunsetsus from B_{i+1} to B_{j-1}.

6 NBs and PBs can govern other bunsetsus, but other types of bunsetsus, like "HIJOUNI(very)" and "SUBETENO(all)," cannot.

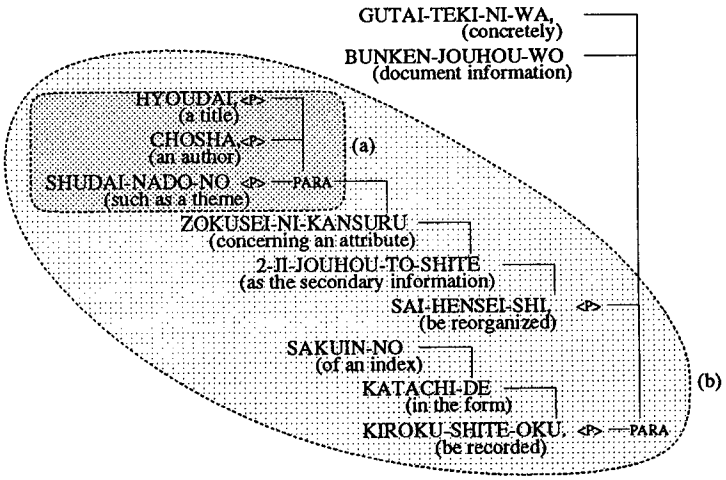


Figure 12
An example of analyzing a long sentence into a dependency structure.

a CS node and assigning each bunsetsu to it (Figure 12a: 'PARA' is the CS node, and the nodes accompanying '<P>' are the root nodes of the dependency trees for conjuncts). Next, the pre- and post-conjuncts [HYOUDAI(*a title*),···SAI-HENSEI-SHI(*be reorganized*),]- [SAKUIN-NO(*of an index*)···KIROKU-SHITE-OKU(*be recorded*).] are analyzed and transformed into dependency trees, and another CS node is created (Figure 12b). Finally, the whole sentence is analyzed, and its dependency tree is obtained.

5.2 Extension of Conjunctive Structures and Recovering Omitted Modifiers

Our method of detecting a CS cannot find where the pre-conjunct begins with complete certainty. For this reason, it is necessary to check whether some modifiers[7] (bunsetsus) to the left of the detected SB can be included in the CS in the stage of dependency analysis. This left-side extension is performed only on CSs containing PBs. This is because modifiers to the left of a CS containing no PB rarely depend on the pre-conjunct alone; usually they depend on the entire CS (this head-dependent relation is handled as the relation to the CS node in the next level analysis) or on a bunsetsu after the CS. When a CS contains PBs, the analysis of its pre-conjunct does not stop at the detected SB, but continues to the bunsetsus to the left of the SB as follows:

- If the bunsetsu depends on a certain bunsetsu apart from the KB in the pre-conjunct, the bunsetsu is regarded as a part of the CS, and the extension operation is continued (Figure 13). Otherwise the extension operation is stopped. The KB is excluded from the candidates for a head, because the head-dependent relation to the KB is handled as the relation to the CS node in the next level analysis.

7 "Modifiers" here mean case components of verbs, too.

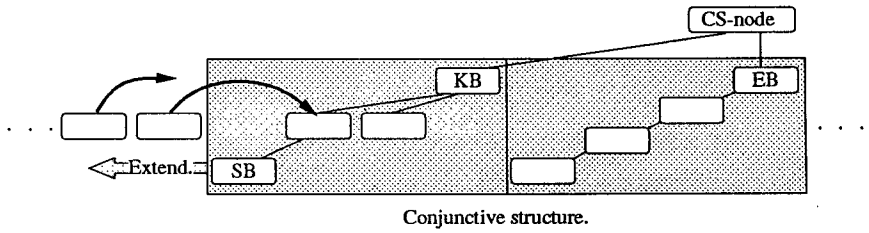


Figure 13
Extension of conjunctive structures.

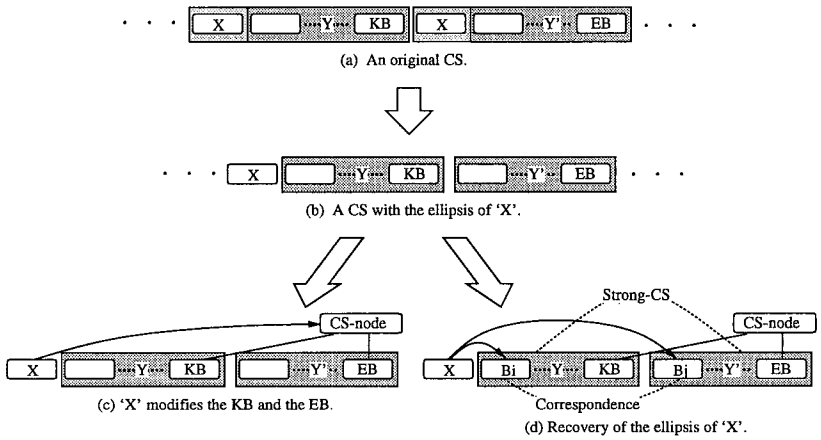


Figure 14
A modifier ellipsis.

- However, if the bunsetsu accompanies the postposition "WA" or a comma, the bunsetsu is not included into the CS and the extension operation is stopped. This is because a bunsetsu of this kind causes a separation in a sentence and usually depends on the entire CS (that is, KB and EB) or another bunsetsu to the right of the CS, not a bunsetsu in the pre-conjunct.

In the sentence in Figure 7, the bunsetsu "SONO(*the*)," which can depend on "KANOUSEI-WO(*possibility*)," is regarded as contained in the CS, but the bunsetsu "KAISHOU-SURU-TAME-NI-WA(*in order to solve*)," which accompanies "WA" and a comma, is not contained in the CS, and the extension of the CS thus ends here.

Through this extension of the CS, the issue of omitted modifiers in a CS can be addressed. When the same modifiers exist in both conjuncts, the modifiers in its post-conjunct are often omitted (Figures 14a and 14b). Among these omitted modifiers, the ones that depend on the EB do not have to be recovered, because a remaining modifier that depends on the KB is treated as depending on the CS node, which means that the

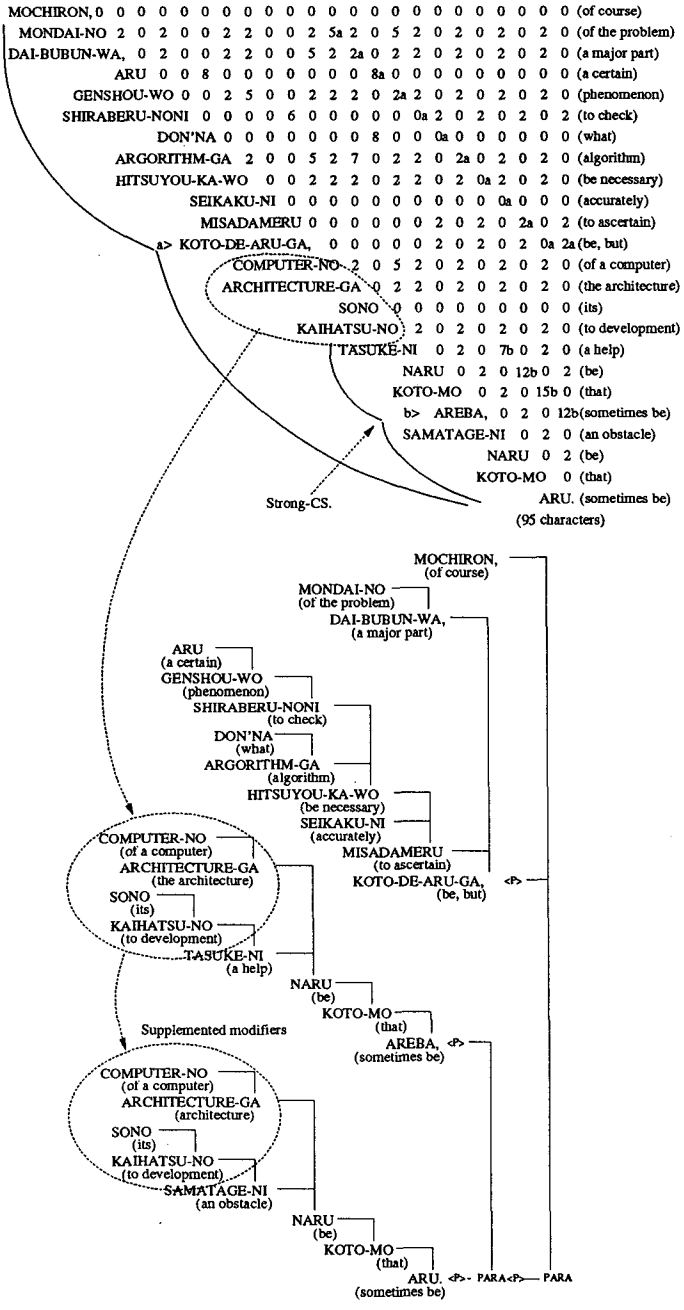


Figure 15
An example of analyzing a long sentence into a dependency structure.

remaining modifier also depends on the EB (Figure 14c). The problem is to recover the omitted modifiers that depend on a bunsetsu in the post-conjunct except the EB. The key point is that Y and Y′ in Figure 14b have a great similarity because they

contain not only similar bunsetsus, KB and EB, but also very similar bunsetsus that originally governed the same modifier X. Therefore, we can detect the possibility of modifier ellipsis by checking the similarity score of the CS obtained when detecting its scope. When the extension operation is performed on the pre-conjunct of a CS that is a *strong CS*, we recover the omitted modifiers by interpreting a bunsetsu that depends on a bunsetsu (B_i) in its pre-conjunct as also depending on the bunsetsu (B_j) in its post-conjunct corresponding to B_i (Figure 14d) (we think B_i corresponds to B_j when the path specifying these conjuncts contains an element $a(i,j)$). A CS that satisfies the following two conditions is called a strong CS:

- The number of bunsetsus in its pre-conjunct (n1) and the number of bunsetsus in its post-conjunct (n2) are about the same, satisfying the equation $(n1/1.3) < n2 < (n1 \times 1.3)$.
- The score of the path specifying the CS is greater than $(n1 + n2) \times 4$.

For example, in the sentence in Figure 15, the detected CS [TASUKE-NI(*a help*)· · · ARE-BA(*sometimes be*),]- [SAMATAGE-NI(*an obstacle*)· · ·ARU(*sometimes be*).] satisfies the above two conditions. Thus, by checking the relation between the CS and the outside modifier phrase "SONO KAIHATSU-NO(*to its development*)" the phrase is considered to depend on both of the bunsetsus "TASUKE-NI(*a help*)" and "SAMATAGE-NI(*an obstacle*)." In the same way, "COMPUTER-NO ARCHITECTURE-GA(*the architecture of a computer*)" is again thought to depend on both the bunsetsu "NARU(*be*)" in the pre-conjunct and the bunsetsu "NARU(*be*)" in the post-conjunct. The dependency tree of this sentence that is supplemented correctly with the omitted modifiers is shown in Figure 15.

5.3 Handling of Analysis Failure and Recovering Omitted Predicates

Another type of ellipsis in CSs that is a serious problem is the omission of predicates in incomplete conjunctive structures. This type of ellipsis can be found by examining the failures of dependency analysis. The failure of dependency analysis here means that a head bunsetsu cannot be found for a certain bunsetsu in a certain range of analysis.

When two predicates in a conjunctive predicative clause are the same, the first predicate is sometimes omitted and the remaining part constitutes the incomplete conjunctive structure (Figures 16a and 16b). In these structures, neither conjunct can be parsed into a dependency tree, because there is no predicate in it that should become the root node of a dependency tree. For this reason, by checking dependency analysis failures, we find incomplete conjunctive structures and start the process of supplementing the CSs with omitted predicates. The conditions for incomplete conjunctive structures are the following (Figure 16c):

- Dependency analysis failure occurs both in the pre- and post-conjuncts.
- The bunsetsus whose heads cannot be found (called FB) contain identical AWs.

The key point is that it is important for successful analysis of CSs containing predicate ellipses to detect the correct scope of the incomplete conjunctive structures. In most cases their scopes can be detected correctly from a significant similarity between the

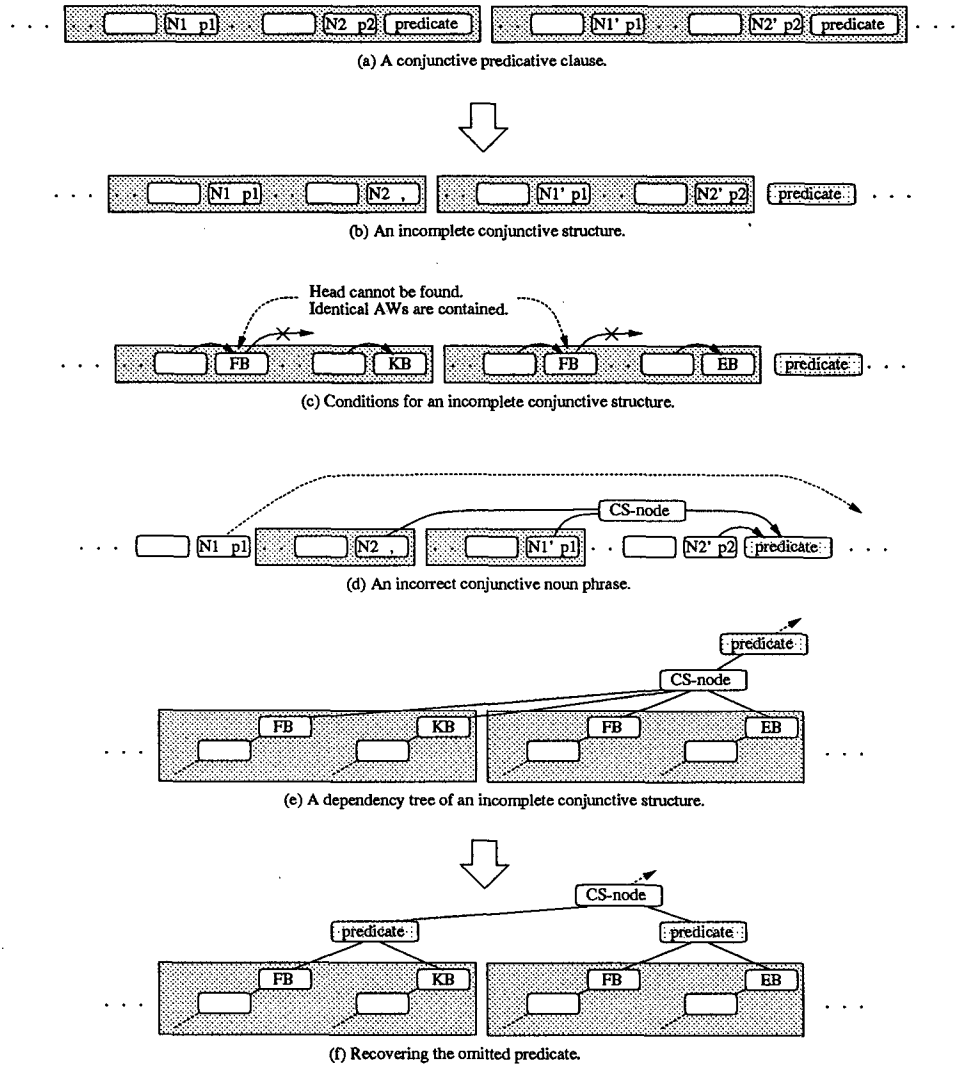


Figure 16
A predicate ellipsis.

pre- and post-conjuncts that contain the case components of the same predicate. That is, the detection of a CS based on the similarity measure smoothly leads to the omitted predicate being recovered. A method that merely searches for the EB as the most similar bunsetsu for the KB might detect an incorrect scope, and in this case the predicate ellipsis cannot be detected, as shown in Figure 16d.

When a CS is regarded as an incomplete conjunctive structure, each series of bunsetsus to the left of an FB is analyzed into a dependency tree, and its root node (FB) is connected to a CS node in addition to the KB and the EB (Figure 16e). When the head of the CS node is found in the *next level analysis*, the head is considered to be the omitted predicate and the dependency tree is transformed by supplementing it with this predicate in the pre-conjunct, as shown in Figure 16f. When the postposition of

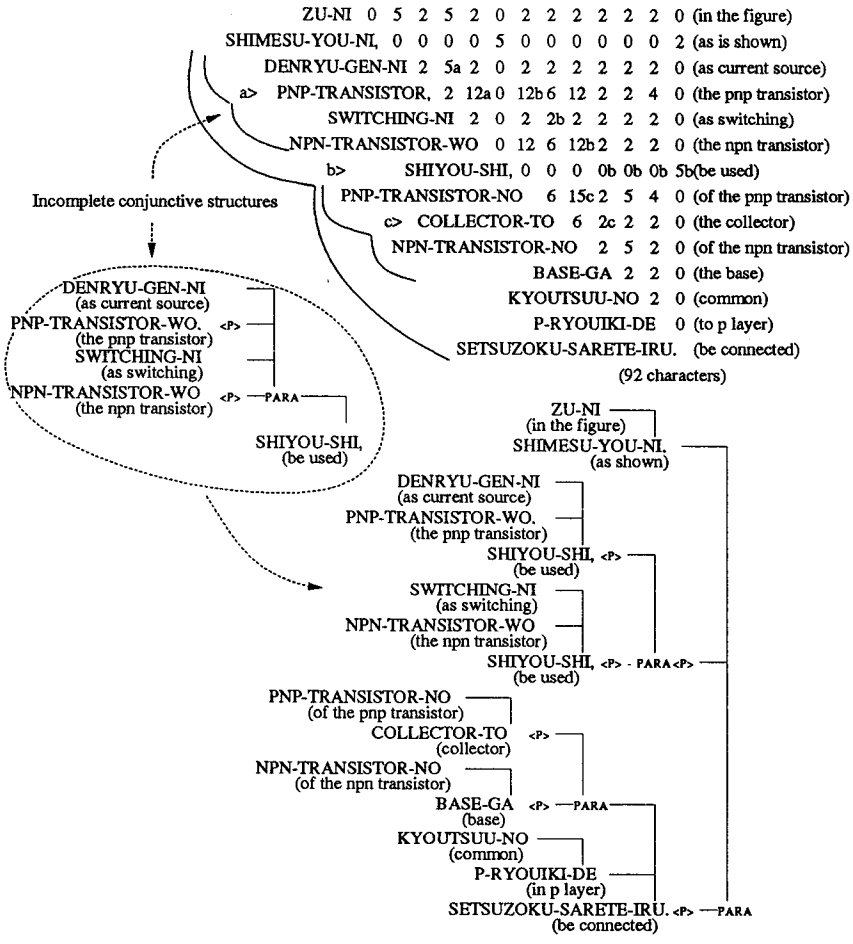


Figure 17
An example of analyzing a long sentence into a dependency structure.

the KB is also omitted (in Figure 16b, p2 is omitted in the KB), the KB is supplemented with the postposition of the EB.

For example, in the sentence in Figure 17, the CS [DENRYU-GEN-NI(*as current source*) PNP-TRANSISTOR(*the pnp transistor*),]- [SWITCHING-NI(*as switching*) NPN-TRANSISTOR-WO(*the npn transistor*)] is recognized as an incomplete conjunctive structure, since the head of the bunsetsu "DENRYU-GEN-NI(*as current source*)" in the pre-conjunct and the bunsetsu "SWITCHING-NI(*as switching*)" in the post-conjunct are not found, and both of them have the same postposition "NI." As a result, FB "DENRYU-GEN-NI(*as current source*)" and FB "SWITCHING-NI(*as switching*)" are connected to the CS node in addition to the KB and EB. In the analysis of the parent CS, it is made clear that this CS node depends on bunsetsu "SHIYOU-SHI(*be used*)," and the dependency tree is transformed by supplementing it with the omitted predicate and the omitted postposition, as shown in Figure 17 (this sentence also contains a conjunc-

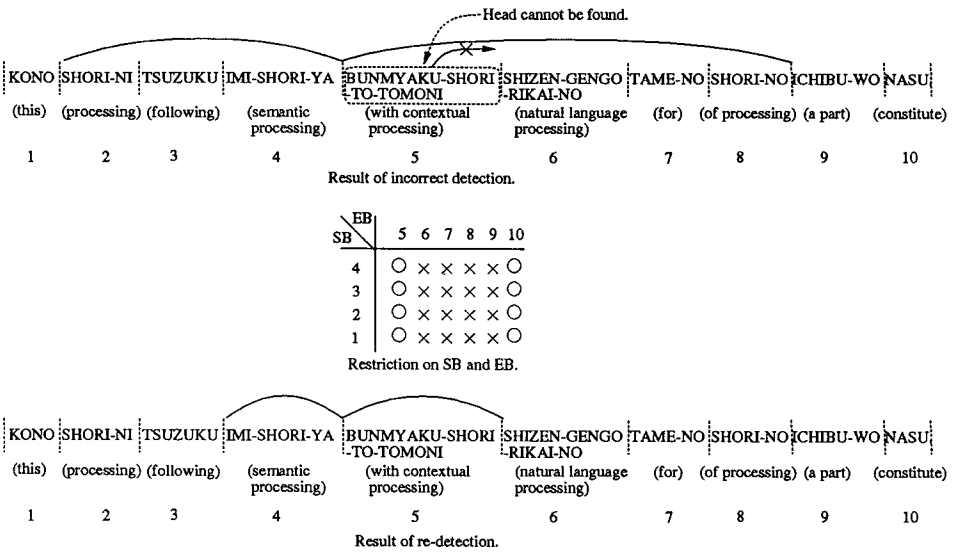


Figure 18
An example of redetecting a conjunctive structure under a failure of analyzing a dependency structure.

tive noun phrase and a conjunctive predicative clause, and all of them are analyzed correctly).

On the other hand, if the dependency analysis of a CS fails and the conditions for incomplete conjunctive structures are not satisfied, we postulate that the detected scope of a CS is incorrect and start the detection of a new CS for the KB. To find a new CS whose pre- and post-conjuncts can be analyzed successfully, the positions of the SB and EB are restricted as follows:

> SB: We examine head-dependent relations in a series of bunsetsus from the first bunsetsu in a sentence to the KB. If there exists a bunsetsu in that range whose head is not found, the analysis must fail for a CS whose pre-conjunct contains this bunsetsu. Therefore, the SB is restricted to be to the right of this bunsetsu.
>
> EB: We examine head-dependent relations in all series of bunsetsus that can be a post-conjunct. If the analysis of a certain series of bunsetsus fails, the last bunsetsu of this series cannot become an EB of a new CS.

After reanalysis of the CS, the analysis returns to the reduction of a sentence by checking the relations between all pairs of CSs. An example of redetecting a CS is shown in Figure 18.

6. Experiments and Discussion

We report the results of analyzing 150 test sentences, which are different from the 30 training sentences used in the parameter adjustment, to illustrate the effectiveness of our method. Test sentences are longer and more complex than sentences in common

usage and consist of 50 sentences composed of 30 to 50 characters, 50 sentences of 50 to 80 characters, and 50 sentences of over 80 characters.[8] All the example sentences shown in this paper belong to these test sentences.

6.1 Experimental Evaluation

We evaluated the results of analyzing 150 Japanese sentences.

First, as shown in Table 4, we classified all the bunsetsus in the 150 sentences into five types: KBs of conjunctive noun phrases, KBs of conjunctive predicative clauses, KBs of incomplete conjunctive structures, bunsetsus that depend on NBs, and bunsetsus that depend on PBs. Then we manually checked these KBs to see whether their corresponding EBs were analyzed correctly; for other bunsetsus, we manually checked whether their heads were analyzed correctly. Table 4 shows a high success ratio for the detection of CSs and a very high success ratio of the dependency analysis on bunsetsu level. These results suggest that the simple heuristic rules for head-dependent relations are good enough to analyze each phrase/clause of the CSs internally and the sentence in which CSs are merged into nodes, respectively.

Second, as shown in the upper part of Table 5, we classified the 150 sentences by their length and according to whether they contain CSs or not. We manually checked whether CSs in each sentence were detected correctly, if they exit, and whether their dependency structures were analyzed correctly. The table shows that CSs are generally well recognized, but the total success ratio of getting proper dependency structures is 65% (97/150). To determine how well a conventional method (described in the introduction) works on such long sentences, we parsed the same test sentences by another method simulating a conventional one. This method uses a simple rule, instead of our dynamic programming method, that a KB depends on the most similar CB (calculated by the process in Section 3.1). It parses a sentence, determining the head bunsetsu from right to left for each bunsetsu in the sentence with this simple rule for CSs, heuristic rules for head-dependent relations (described in Section 5.1), and the no-cross condition. The result of this method (the lower part of Table 5) clearly shows the superiority of our method over the conventional method.

Third, we report the results of the redetection of CSs and the recovery of omitted components.

- The redetection of CSs was activated only for incorrect CSs, so we can conclude that the conditions for performing redetection are reasonable. Out of 215 CSs, 180 were obtained correctly by the first CS detection (the success ratio is 84%). Five CSs were redetected because of incorrect relation to other CSs, and all of them were analyzed correctly. Eight CSs were redetected because of the failure in obtaining a dependency structure, and five out of them were recognized correctly. Finally, 190 CSs out of 215 were obtained correctly (the success ratio is 88%).
- Eleven out of 215 detected CSs satisfied the conditions for a strong CS. One strong CS was an incorrectly detected CS, and this problem is mentioned in the following section. For two of the ten correctly detected strong CSs, the omitted components that depend on one of the bunsetsus

8 Test sentences are collected at random from the following three sources: 30 sentences from the *Encyclopedic Dictionary of Computer Science* (published by Iwanami Publishing Co.); 60 sentences from abstracts of papers from *Japan Information Center of Science and Technology* and 60 sentences from the popular science journal *Science*, translated into Japanese (Volume 17, number 12, "Advanced computing for science").

Table 4
Analysis result on bunsetsu level

	a	b	c
KBs of conjunctive noun phrases	127	113	89%
KBs of conjunctive predicative clauses	86	75	87%
KBs of incomplete conjunctive structures	2	2	100%
Total of KBs	215	190	88%
Bunsetsus depending on NBs	765	744	97%
Bunsetsus depending on PBs	971	941	97%
Total of bunsetsus other than KBs	1736	1685	97%

a: The number of bunsetsus that were classified into this category.
b: The number of bunsetsus whose corresponding EBs were determined correctly, or the number of bunsetsus whose heads were determined correctly.
c: Success ratio.

Table 5
Analysis result on sentence level

Number of characters in a sentence	30–50			50–80			80–149			Total		
Our method	a	b	c	a	b	c	a	b	c	a	b	c
Sentences with no CS	29	—	25	10	—	5	4	—	3	43	—	33
Sentences with CSs	21	15	14	40	34	30	46	37	20	107	86	64
Total	50	—	39	50	—	35	50	—	23	150	—	97
Conventional method	a	b	c	a	b	c	a	b	c	a	b	c
Sentences with no CS	29	—	25	10	—	5	4	—	3	43	—	33
Sentences with CSs	21	14	12	40	27	22	46	19	7	107	60	41
Total	50	—	37	50	—	27	50	—	10	150	—	74

a: The number of sentences that were classified into this category.
b: The number of sentences in which all the CSs were detected correctly.
c: The number of sentences whose whole dependency structures were analyzed correctly.

in the post-conjunct other than the EB (the case of Figure 14d) were recovered correctly. There was no modifier ellipsis of this type that could not be found by our method in the test sentences. Other strong CSs had omitted modifiers depending on the EB (the case of Figure 14c), or had no omitted modifiers.

- There were two incomplete conjunctive structures in the test sentences. Both of them were found by our method, and the omitted predicates concerning them were recovered correctly.

We analyzed sentences of considerable length, consisting of many bunsetsus (the average number of bunsetsus in a sentence was 14.3). There are many candidate heads for each bunsetsu in such a sentence, making the possibility for incorrect head-dependent relations in the dependency structure of a sentence significant. Considering these conditions and comparing results using our method with those using the conventional method, the total success ratio for determining correct dependency structures

Table 6
Examples of failure of analysis

(i) JISSAI (*in fact*), CHOSHA-TACHI-WA (*authors*) {[KORE-WO (*it*) TSUKATTE (*using*), JUURYOKU-SOUGO-SAYOU-GA (*gravitationally interacting*) SHIHAI-SURU (*governing*)] TENTAI-NO (*astronomical*) UNDOU-NI-TSUITE (*about the motion*), KOUSEIDO-DE (*high-precision*) KOUSOKU-NO (*high-speed*) SUUCHI-KEISAN-GA (*numerical computation*) DEKIRU (*can*) DIGITAL-ORRERY-TO-IU (*called Digital Orrery*) SENYOU-COMPUTER-WO (*special-purpose computer*) SEISAKU-SHITE-IRU (*create*).}

(ii) ... {HYOUGEN-GOTO-NI (*for every expression*) YOUI-SHITA (*prepared*) [KETSUGOU-KA-KOUZOU-CHUU-NO (*in a combinative structure*) KETSUGOU-KA-YOUSO-TO (*combinative elements*) BUN-CHUU-NO (*in a sentence*) KAKU-YOUSO-NO (*between case elements*)]} TAIOU-WO (*correspondence*) ...

(iii) KORERA (*these*) KAISEKI-SHUHOU-NO (*of analysis methods*) KYOUTSUU-SHITA (*common*) MONDAI-TO-SHITE (*as problems*) BUNPOU-KISOKU-GA (*grammar rules*) OOKIKU-NATTA (*increasing*) BAAI-NO (*in the case*) [KISOKU-NO (*of rules*) { KAKUCHOU-YA (*and extension*) HOSHU-NO (*of maintenance*)} KONNAN-GA (*difficulty*)] AGE-RARERU (*can be thought*).

(iv) ... NIHONGO-TAIWA-BUN-KAISEKI-BU-HA (*Japanese dialogue analytic module*), {[KAISEKI-KATEI-NO (*of the analysis process*) SEIGYO-GA (*control*) JIYUUNA (*be free*) ACTIVE-CHART-KAISEKI-HOU-TO (*and Active Chart Parsing*) TAN'ITSUKA-NI (*on unification*) MOTOZUITA (*based*) GOI-TOUJI-TEKINA (*lexicon based*) BUNPOU-TEKI-WAKUGUMI-DEARU (*being the grammatical framework*)] HPSG-WO (*HPSG*)} SAIYOU-SHITE-IRU (*be adopted*).

(v) ... {[HIBUN-NI-TAISURU (*for illegal sentences*) TEISHI-SEI-YA (*and termination*) SHUTSURYOKU-SURU (*outputted*) BUN-NO (*of sentences*) AIMAISA-NO (*of ambiguities*)] JOUGEN-NI-TSUITE (*about the maximum*)} HOSHOU-GA-NAI (*there is no guarantee*).

for a complete sentence, 65%, can be considered to be fairly good. Although one-third of the dependency structures after this analysis process included some errors, their major structures, that is, their conjunctive structures and basic dependency structures, were detected correctly in most cases. This can be seen from the high scores in Table 4.

6.2 Discussion of Incorrect Analysis

It is possible to classify some of the causes of incorrect analyses arising from our method. Table 6 gives some examples of errors in recognizing CSs. Here the underlined bunsetsus are KBs. The incorrectly calculated scope of a CS is enclosed by square brackets, and the correct scope is enclosed by curly brackets.

- Our assumption that both conjuncts contain about the same number of bunsetsus is useful in detecting most CSs. Even if the number of bunsetsus of two conjuncts is somewhat different, a correct CS can be obtained with the help of the penalty points, which reduces the possibility that a CS contains high SL bunsetsus, and with the extension of the pre-conjunct, and so on. However, it is difficult to recognize a CS that is extremely unbalanced. In sentence (i) in Table 6, the KB "TSUKATTE(*using*)" in the beginning part of the sentence should correspond to the last CB "SEISAKU-SHITE-IRU(*create*)." and a short clause "KORE-WO TSUKATTE(*using it*)" corresponds to the following

long clause. Since this CS is extremely unbalanced, an incorrect CS enclosed by the square brackets, which is relatively well balanced, is detected. To detect the CS correctly, the causal relation between "TSUKATTE(*using*)" and "SEISAKU-SHITE-IRU(*create*)." will have to be taken into consideration.

- It is very difficult to recognize a CS in which the last part of the pre-conjunct is very similar to the whole post-conjunct, and there are additional modifiers only in the pre-conjunct, such as ... A B C, B' C'..., in which A B C and B' C' are conjoined. In fact, A depends on either B or C. However, our method detects the incorrect strong CS ([B C]-[B' C']) and consider A as an omitted modifier, so that A is interpreted as depending on both B and B' or both C and C'. Sentence (ii) is an example of this case. Since, in most cases, modifiers to the left of strong CS depend on both conjuncts, it is very difficult to analyze such expressions.

- In detecting CSs it is essential to give an appropriate similarity value between bunsetsus. More precise similarity measures will be necessary to improve the success ratio of detecting CSs significantly. One way to provide this is to make a finer distinction when scoring part-of-speech similarity (e.g., nouns can be divided into numerals, proper nouns, common nouns, and action nouns that become verbs by the combination with "SURU(*do*)"). For instance, sentence (iii) will be analyzed correctly if the similarity score between the action noun "KAKUCHOU(*extension*)" and the action noun "HOSHU(*maintenance*)" is greater than that between the action noun "KAKUCHOU(*extension*)" and the common noun "KONNAN(*difficulty*)." Another way is to give similarity scores for technical terms by computing the distance measure between them in a technical term thesaurus. At present, the similarity between compound technical terms is detected only by partial character matching between them. If points were given to the similarity between "ACTIVE-CHART-KAISEKI-HOU(*Active Chart Parsing*)" and "HPSG(*Head-driven Phrase Structure Grammar*)" using a thesaurus (both terms are close in the sense of grammar and parsing), sentence (iv) would be analyzed correctly.

- Some of the sentences that were analyzed incorrectly are too subtle even for a human to find the CSs correctly. Only specialists in the subject of the text can interpret the structure correctly with the help of their expert knowledge. The CS in sentence (v) cannot be detected correctly without expert knowledge of the subject. Such expressions are, however, very few in actual texts.

- Our heuristic rules for dependency analysis are simple and do not handle some difficult or subtle expressions. The analysis of such expressions is the target of another line of research in itself. One major problem is the structural ambiguity in the succession of NBs, such as "A-NO B-NO C." While our rules now simply conclude that "A-NO" depends on "B," sometimes "A-NO" depends on "C."

7. Concluding Remarks

We have shown that a variety of conjunctive structures in Japanese sentences can be detected using a certain similarity measure and that information about conjunctive structures enables the syntactic analysis to be more robust and successful in handling long and complex sentences. There are still some expressions that cannot be recognized by the proposed method, and one might hasten to rely on semantic information in the hope of getting proper analyses for these remaining cases. Semantic information, however, is not as reliable as syntactic information, and we have to make further efforts to find some syntactic rather than semantic relations even in these difficult cases. Phrase structure grammar or other existing grammar formalisms may not be applicable in detecting the subtle syntactic relations among several words in a sentence. We have to find new methods to detect them. To make further progress in this field, we feel it is necessary to be able to take into consideration more possible interactions among a wider range of components of long sentences.

References

Agarwal, R., and Boggess, L. (1992). "A simple but useful approach to conjunct identification." In *Proceedings, 30th Annual Meeting of Association for Computational Linguistics*, 15–21.

Dahl, V., and McCord, M. C. (1983). "Treating coordination in logic grammars." *American Journal of Computational Linguistics* 9(2):69–91.

Fong, S., and Berwick, R. C. (1985). "New approach to parsing conjunctions using Prolog." In *Proceedings, 23th Annual Meeting of Association for Computational Linguistics*, 118–126.

Hirschman, L. (1986). "Conjunction in meta-restriction grammar." *Journal of Logic Programming* 3(4):299–328.

Nagao, M.; Tsujii, J.; Tanaka, N.; and Ishikawa, M. (1983). "Conjunctive phrases in scientific and technical papers and their analysis (in Japanese)." *Information Processing Society of Japan SIG Reports*, NL-36.

National Language Research Institute (1964). *Bunrui Goi Hyou (Word List by Semantic Principles)*. Shuuei Publishing.

Kaplan, R. M., and Maxwell, J. T. (1988). "Constituent coordination in Lexical-Functional Grammar." In *Proceedings, Eighth International Conference on Computational Linguistics* 1:303–305.

Sag, I. A., et al. (1985). "Coordination and how to distinguish categories." *Natural Language and Linguistic Theory* 3:117–171.

Sedogbo, C. (1985). "A meta-grammar for handling coordination in logic grammars." In *Natural Language Understanding and Logic Programming*, edited by V. Dahl and P. Saint-Dizier, 65–78. Amsterdam: North Holland.

Shudo, K.; Yoshimura, K.; and Tsuda, K. (1986). "Coordinate structures in Japanese technical sentences (in Japanese)." *Transactions of Information Processing Society of Japan* 27(2):183–190.

Steedman, M. J. (1990). "Gapping as constituent coordination." *Linguistics and Philosophy* 13:207–263.

Woods, W. A. (1973). "An experimental parsing system for transition network grammars." In *Natural Language Processing*, edited by R. Rustin, 111–154. Algorithmics Press.