# Analyzing Dependencies of Japanese Subordinate Clauses based on Statistics of Scope Embedding Preference

**Takehito Utsuro, Shigeyuki Nishiokayama, Masakazu Fujio, Yuji Matsumoto**
**Graduate School of Information Science, Nara Institute of Science and Technology**
**8916-5, Takayama-cho, Ikoma-shi, Nara, 630-0101, JAPAN**
E-mail: *utsuro@is.aist-nara.ac.jp*, URL: *http://cl.aist-nara.ac.jp/~utsuro/*

## Abstract

This paper proposes a statistical method for learning dependency preference of Japanese subordinate clauses, in which scope embedding preference of subordinate clauses is exploited as a useful information source for disambiguating dependencies between subordinate clauses. Estimated dependencies of subordinate clauses successfully increase the precision of an existing statistical dependency analyzer.

## 1 Introduction

In the Japanese language, since word order in a sentence is relatively free compared with European languages, dependency analysis has been shown to be practical and effective in both rule-based and stochastic approaches to syntactic analysis. In dependency analysis of a Japanese sentence, among various source of ambiguities in a sentence, dependency ambiguities of subordinate clauses are one of the most problematic ones, partly because word order in a sentence is relatively free. In general, dependency ambiguities of subordinate clauses cause scope ambiguities of subordinate clauses, which result in enormous number of syntactic ambiguities of other types of phrases such as noun phrases.[1]

In the Japanese linguistics, a theory of Minami (1974) regarding scope embedding preference of subordinate clauses is well-known. Minami (1974) classifies Japanese subordinate clauses according to the breadths of their scopes and claim that subordinate clauses which inherently have narrower scopes are embedded within the scopes of subordinate clauses which inherently have broader scopes (details are in section 2). By manually analyzing several raw corpora, Minami (1974) classifies various types of Japanese subordinate clauses into three categories, which are totally ordered by the embedding relation of their scopes. In the Japanese computational linguistics community, Shirai et al. (1995) employed Minami (1974)'s theory on scope embedding preference of Japanese subordinate clauses and applied it to rule-based Japanese dependency analysis. However, in their approach, since categories of subordinate clauses are obtained by manually analyzing a small number of sentences, their coverage against a large corpus such as EDR bracketed corpus (EDR, 1995) is quite low.[2]

In order to realize a broad coverage and high performance dependency analysis of Japanese sentences which exploits scope embedding preference of subordinate clauses, we propose a corpus-based and statistical alternative to the rule-based manual approach (section 3).[3]

[1]In our preliminary corpus analysis using the stochastic dependency analyzer of Fujio and Matsumoto (1998), about 30% of the 210,000 sentences in EDR bracketed corpus (EDR, 1995) have dependency ambiguities of subordinate clauses, for which the precision of chunk (bunsetsu) level dependencies is about 85.3% and that of sentence level is about 25.4% (for best one) ~ 35.8% (for best five), while for the rest 70% of EDR bracketed corpus, the precision of chunk (bunsetsu) level dependencies is about 86.7% and that of sentence level is about 47.5% (for best one) ~ 60.2% (for best five). In addition to that, when assuming that those ambiguities of subordinate clause dependencies are initially resolved in some way, the chunk level precision increases to 90.4%, and the sentence level precision to 40.6% (for best one) ~ 67.7% (for best five). This result of our preliminary analysis clearly shows that dependency ambiguities of subordinate clauses are among the most problematic source of syntactic ambiguities in a Japanese sentence.

[2]In our implementation, the coverage of the categories of Shirai et al. (1995) is only 30% for all the subordinate clauses included in the whole EDR corpus.

[3]Previous works on statistical dependency analysis include Fujio and Matsumoto (1998) and Haruno et al. (1998) in Japanese analysis as well as Lafferty et al. (1992), Eisner (1996), and Collins (1996) in English analysis. In later sections, we discuss the advantages of our approach over several closely related previous works.

Table 1: Word Segmentation, POS tagging, and *Bunsetsu* Segmentation of A Japanese Sentence

| Word Segmentation | Tenki | ga | yoi | kara | dekakeyou |
|---|---|---|---|---|---|
| POS (+ conjugation form) Tagging | noun | case-particle | adjective (base) | predicate-conjunctive-particle | verb (volitional) |
| Bunsetsu Segmentation (Chunking) | Tenki-ga | | yoi-kara | | dekakeyou |
| English Translation | *weather* | subject | *fine* | *because* | *let's go out* |
| | *(Because the weather is fine, let's go out.)* | | | | |

First, we formalize the problem of deciding scope embedding preference as a classification problem, in which various types of linguistic information of each subordinate clause are encoded as features and used for deciding which one of given two subordinate clauses has a broader scope than the other. As in the case of Shirai et al. (1995), we formalize the problem of deciding dependency preference of subordinate clauses by utilizing the correlation of scope embedding preference and dependency preference of Japanese subordinate clauses. Then, as a statistical learning method, we employ the decision list learning method of Yarowsky (1994), where optimal combination of those features are selected and sorted in the form of decision rules, according to the strength of correlation between those features and the dependency preference of the two subordinate clauses. We evaluate the proposed method through the experiment on learning dependency preference of Japanese subordinate clauses from the EDR bracketed corpus (section 4). We show that the proposed method outperforms other related methods/models. We also evaluate the estimated dependencies of subordinate clauses in Fujio and Matsumoto (1998)'s framework of the statistical dependency analysis of a whole sentence, in which we successfully increase the precisions of both chunk level and sentence level dependencies thanks to the estimated dependencies of subordinate clauses.

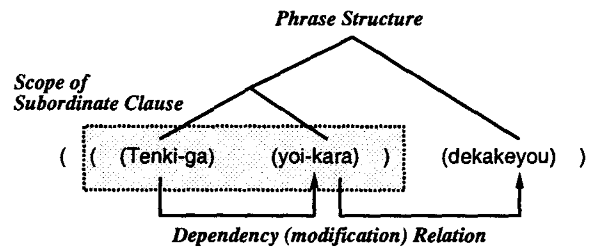



Figure 1: An Example of Japanese Subordinate Clause (taken from the Sentence of Table 1)

## 2 Analyzing Dependencies between Japanese Subordinate Clauses based on Scope Embedding Preference

### 2.1 Dependency Analysis of A Japanese Sentence

First, we overview dependency analysis of a Japanese sentence. Since words in a Japanese sentence are not segmented by explicit delimiters, input sentences are first word segmented, part-of-speech tagged, and then chunked into a sequence of segments called *bunsetsus*.[4] Each chunk (bunsetsu) generally consists of a set of content words and function words. Then, dependency relations among those chunks are estimated, where most practical dependency analyzers for the Japanese language usually assume the following two constraints:

1. Every chunk (bunsetsu) except the last one modifies only one posterior chunk (bunsetsu).

2. No modification crosses to other modifications in a sentence.

Table 1 gives an example of word segmentation, part-of-speech tagging, and *bunsetsu* segmentation (chunking) of a Japanese sentence, where the verb and the adjective are tagged with their parts-of-speech as well as conjugation forms. Figure 1 shows the phrase structure, the bracketing,[5] and the dependency (modification) relation of the chunks (bunsetsus) within the sentence.

[4]Word segmentation and part-of-speech tagging are performed by the Japanese morphological analyzer Chasen (Matsumoto et al., 1997), and chunking is done by the preprocessor used in Fujio and Matsumoto (1998).

[5]The phrase structure and the bracketing are shown just for explanation, and we do not consider them but consider only dependency relations in the analysis throughout this paper.

A Japanese *subordinate clause* is a clause whose head chunk satisfies the following properties.

1. The content words part of the chunk (bunsetsu) is one of the following types:
   (a) A predicate (i.e., a verb or an adjective).
   (b) nouns and a copula like "$Noun_1$ *dearu*" (in English, "*be* $Noun_1$").
2. The function words part of the chunk (bunsetsu) is one of the following types:
   (a) Null.
   (b) Adverb type such as "$Verb_1$ *ippou-de*" (in English, "(subject) $Verb_1$ ..., *on the other hand,*").
   (c) Adverbial noun type such as "$Verb_1$ *tame*" (in English, "*in order to* $Verb_1$").
   (d) Formal noun type such as "$Verb_1$ *koto*" (in English, gerund "$Verb_1$-ing").
   (e) Temporal noun type such as "$Verb_1$ *mae*" (in English, "*before* (subject) $Verb_1$ ...").
   (f) A predicate conjunctive particle such as "$Verb_1$ *ga*" (in English, "although (subject) $Verb_1$ ...,").
   (g) A quoting particle such as "$Verb_1$ *to (iu)*" (in English, "*(say) that* (subject) $Verb_1$ ...").
   (h) (a)~(g) followed by topic marking particles and/or sentence-final particles.

Figure 2: Definition of Japanese Subordinate Clause

### 2.2 Japanese Subordinate Clause

The following gives the definition of what we call a "Japanese *subordinate clause*" throughout this paper. A *clause* in a sentence is represented as a sequence of chunks. Since the Japanese language is a head-final language, the clause head is the final chunk in the sequence. A grammatical definition of a Japanese *subordinate clause* is given in Figure 2.[6] For example, the Japanese sentence in Table 1 has one subordinate clause, whose scope is indicated as the shaded rectangle in Figure 1.

### 2.3 Scope Embedding Preference of Subordinate Clauses

We introduce the concept of Minami (1974)'s classification of Japanese subordinate clauses by describing the more specific classification by Shirai et al. (1995). From 972 newspaper summary sentences, Shirai et al. (1995) manually extracted 54 clause final function words of Japanese subordinate clauses and classified them into the following three categories according to the embedding relation of their scopes.

**Category A:** Seven expressions representing simultaneous occurrences such as "$Verb_1$ *to-tomoni* ($Clause_2$)" and "$Verb_1$ *nagara* ($Clause_2$)".

**Category B:** 46 expressions representing cause and discontinuity such as "$Verb_1$ *te* ($Clause_2$)" (in English "$Verb_1$ *and* ($Clause_2$)") and "$Verb_1$ *node*" (in English "*because* (subject) $Verb_1$ ...,").

**Category C:** One expression representing independence, "$Verb_1$ *ga*" (in English, "although (subject) $Verb_1$ ...,").

The category A has the narrowest scope, while the category C has the broadest scope, i.e.,

$$\text{Category A} \prec \text{Category B} \prec \text{Category C}$$

where the relation '$\prec$' denotes the embedding relation of scopes of subordinate clauses. Then, scope embedding preference of Japanese subordinate clauses can be stated as below:

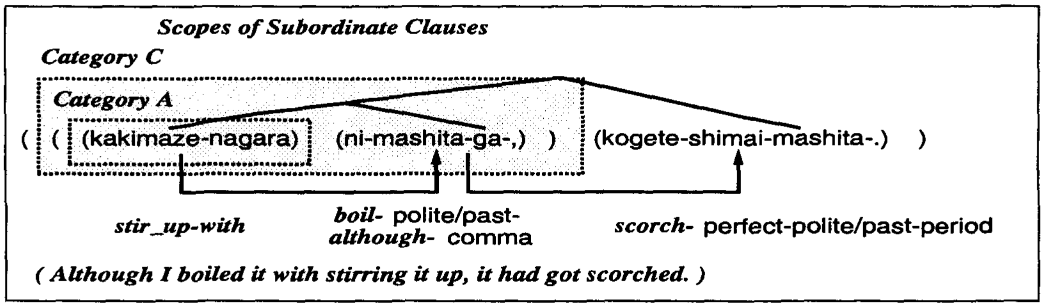
**Scope Embedding Preference of Japanese Subordinate Clauses**

1. A subordinate clause can be embedded within the scope of another subordinate clause which inherently has a scope of the same or a broader breadth.
2. A subordinate clause can not be embedded within the scope of another subordinate clause which inherently has a narrower scope.

For example, a subordinate clause of 'Category B' can be embedded within the scope of another subordinate clause of 'Category B' or 'Category C', but not within that of 'Category A'. Figure 3

[6]This definition includes adnominal or noun phrase modifying clauses "$Clause_1$ ($NP_1$)" (in English, relative clauses "($NP_1$) that $Clause_1$"). Since an adnominal clause does not modify any posterior subordinate clauses, but modifies a posterior noun phrase, we regard adnominal clauses only as modifees when considering dependencies between subordinate clauses.

(a) Category A $\prec$ Category C




(b) Category C $\succ$ Category A

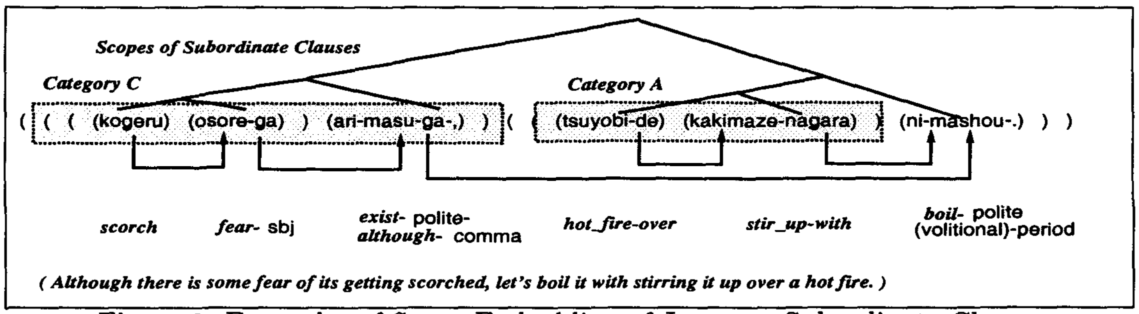



Figure 3: Examples of Scope Embedding of Japanese Subordinate Clauses

(a) gives an example of an anterior Japanese subordinate clause (*"kakimaze-nagara"*, Category A), which is embedded within the scope of a posterior one with a broader scope (*"ni-mashita-ga-,"*, Category C). Since the posterior subordinate clause inherently has a broader scope than the anterior, the anterior is embedded within the scope of the posterior. On the other hand, Figure 3 (b) gives an example of an anterior Japanese subordinate clause (*"ari-masu-ga-,"*, Category C), which is not embedded within the scope of a posterior one with a narrower scope (*"kakimaze-nagara"*, Category A). Since the posterior subordinate clause inherently has a narrower scope than the anterior, the anterior is not embedded within the scope of the posterior.

### 2.4 Preference of Dependencies between Subordinate Clauses based on Scope Embedding Preference

Following the scope embedding preference of Japanese subordinate clauses proposed by Minami (1974), Shirai et al. (1995) applied it to rule-based Japanese dependency analysis, and proposed the following preference of deciding dependencies between subordinate clauses. Suppose that a sentence has two subordinate clauses $Clause_1$ and $Clause_2$, where the head vp chunk of $Clause_1$ precedes that of $Clause_2$.

**Dependency Preference of Japanese Subordinate Clauses**

1. The head vp chunk of $Clause_1$ can modify that of $Clause_2$ if $Clause_2$ inherently has a scope of the same or a broader breadth compared with that of $Clause_1$.
2. The head vp chunk of $Clause_1$ can not modify that of $Clause_2$ if $Clause_2$ inherently has a narrower scope compared with that of $Clause_1$.

## 3 Learning Dependency Preference of Japanese Subordinate Clauses

As we mentioned in section 1, the rule-based approach of Shirai et al. (1995) to analyzing dependencies of subordinate clauses using scope embedding preference has serious limitation in its coverage against corpora of large size for practical use. In order to overcome the limitation of the rule-based approach, in this section, we propose a method of learning dependency preference of Japanese subordinate clauses from a bracketed corpus. We formalize the problem of deciding scope embedding preference as a classification problem, in which various types of linguistic information of each subordinate clause are encoded as features and used for deciding which one of given two subordinate clauses has a broader scope than the other. As a statistical learning method, we employ the decision list learning method of Yarowsky (1994).

Table 2: Features of Japanese Subordinate Clauses

| Feature Type | # of Features | Each Binary Feature |
|---|---|---|
| Punctuation | 2 | with-comma, without-comma |
| Grammatical (some features have distinction of chunk-final/middle) | 17 | adverb, adverbial-noun, formal-noun, temporal-noun, quoting-particle, copula, predicate-conjunctive-particle, topic-marking-particle, sentence-final-particle |
| Conjugation form of chunk-final conjugative word | 12 | stem, base, mizen, ren'you, rentai, conditional, imperative, *ta*, *tari*, *te*, conjecture, volitional |
| Lexical (lexicalized forms of 'Grammatical' features, with more than 9 occurrences in EDR corpus) | 235 | adverb (e.g., *ippou-de, irai*), adverbial-noun (e.g., *tame, baai*) topic-marking-particle (e.g., *ha, mo*), quoting-particle (*to*), predicate-conjunctive-particle (e.g., *ga, kara*), temporal-noun (e.g., *ima, shunkan*), formal-noun (e.g., *koto*), copula (*dearu*), sentence-final-particle (e.g., *ka, yo*) |

### 3.1 The Task Definition

Considering the dependency preference of Japanese subordinate clauses described in section 2.4, the following gives the definition of our task of deciding the dependency of Japanese subordinate clauses. Suppose that a sentence has two subordinate clauses $Clause_1$ and $Clause_2$, where the head vp chunk of $Clause_1$ precedes that of $Clause_2$. Then, our task of deciding the dependency of Japanese subordinate clauses is to distinguish the following two cases:

1. The head vp chunk of $Clause_1$ modifies that of $Clause_2$.
2. The head vp chunk of $Clause_1$ does not modify that of $Clause_2$, but modifies that of another subordinate clause or the matrix clause which *follows* $Clause_2$.

Roughly speaking, the first corresponds to the case where $Clause_2$ inherently has a scope of the same or a broader breadth compared with that of $Clause_1$, while the second corresponds to the case where $Clause_2$ inherently has a narrower scope compared with that of $Clause_1$.[7]

### 3.2 Decision List Learning

A decision list (Yarowsky, 1994) is a sorted list of the decision rules each of which decides the value of a *decision* $D$ given some *evidence* $E$. Each decision rule in a decision list is sorted in descending order with respect to some preference value, and rules with higher preference values are applied first when applying the decision list to some new test data.

First, let the random variable $D$ representing a decision varies over several possible values, and the random variable $E$ representing some evidence varies over '1' and '0' (where '1' denotes the presence of the corresponding piece of evidence, '0' its absence). Then, given some training data in which the correct value of the decision $D$ is annotated to each instance, the conditional probabilities $P(D=x \mid E=1)$ of observing the decision $D=x$ under the condition of the presence of the evidence $E$ ($E=1$) are calculated and the decision list is constructed by the following procedure.

1. For each piece of evidence, calculate the *likelihood ratio* of the conditional probability of a decision $D=x_1$ (given the presence of that piece of evidence) to the conditional probability of the rest of the decisions $D=\neg x_1$:

$$\log_2 \frac{P(D=x_1 \mid E=1)}{P(D=\neg x_1 \mid E=1)}$$

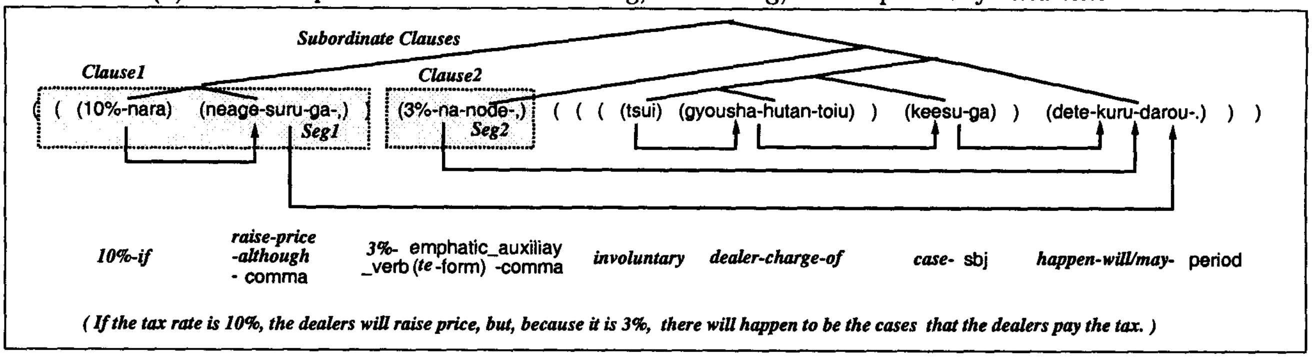
   Then, a decision list is constructed with pieces of evidence sorted in descending order with respect to their likelihood ratios.[8]

2. The final line of a decision list is defined as 'a default', where the likelihood ratio is calculated as the ratio of the largest marginal probability of the decision $D=x_1$ to the marginal proba-

[7] Our modeling is slightly different from those of other standard approaches to statistical dependency analysis (Collins, 1996; Fujio and Matsumoto, 1998; Haruno et al., 1998) which simply distinguish the two cases: the case where dependency relation holds between the given two vp chunks or clauses, and the case where dependency relation does not hold. In contrast to those standard approaches, we ignore the case where the head vp chunk of $Clause_1$ modifies that of another subordinate clause which *precedes* $Clause_2$. This is because we assume that this case is more loosely related to the scope embedding preference of subordinate clauses.

[8] Yarowsky (1994) discusses several techniques for avoiding the problems which arise when an observed count is 0. Among those techniques, we employ the simplest one, i.e., adding a small constant $\alpha$ ($0.1 \leq \alpha \leq 0.25$) to the numerator and denominator. With this modification, more frequent evidence is preferred when there exist several evidences for each of which the conditional probability $P(D=x \mid E=1)$ equals to 1.

(a) An Example Sentence with Chunking, Bracketing, and Dependency Relations




(b) Feature Expression of Head VP Chunk of Subordinate Clauses

| Head VP Chunk of Subordinate Clause | Feature Set |
|---|---|
| $Seg_1$ : "neage-suru-ga-," | $\mathcal{F}_1 = \{$ with-comma, predicate-conjunctive-particle(chunk-final), predicate-conjunctive-particle(chunk-final)-"*ga*" $\}$ |
| $Seg_2$ : "3%-na-node-," | $\mathcal{F}_2 = \{$ with-comma, chunk-final-conjugative-word-*te*-form $\}$ |

(c) Evidence-Decision Pairs for Decision List Learning

| Evidence $E$ ($E=1$) (feature names are abbreviated) | | |
|---|---|---|
| $\mathcal{F}_1$ | $\mathcal{F}_2$ | Decision $D$ |
| with-comma | with-comma | "beyond" |
| with-comma | *te*-form | "beyond" |
| with-comma | with-comma, *te*-form | "beyond" |
| pred-conj-particle(final) | with-comma | "beyond" |
| ... | ... | ... |
| with-comma, pred-conj-particle(final) | with-comma | "beyond" |
| ... | ... | ... |
| pred-conj-particle(final)-"*ga*" | with-comma | "beyond" |
| ... | ... | ... |
| with-comma, pred-conj-particle(final)-"*ga*" | with-comma | "beyond" |
| ... | ... | ... |

Figure 4: An Example of Evidence-Decision Pair of Japanese Subordinate Clauses

bility of the rest of the decisions $D = \neg x_1$:

$$\log_2 \frac{P(D=x_1)}{P(D=\neg x_1)}$$

The 'default' decision of this final line is $D=x_1$ with the largest marginal probability.

### 3.3 Feature of Subordinate Clauses

Japanese subordinate clauses defined in section 2.2 are encoded using the following four types of features: i) Punctuation: represents whether the head vp chunk of the subordinate clause is marked with a comma or not, ii) Grammatical: represents parts-of-speech of function words of the head vp chunk of the subordinate clause,[9] iii) Conjugation form of chunk-final conjugative word: used when the chunk-final word is conjugative, iv) Lexical: lexicalized forms of 'Grammatical' features which appear more than 9 times in EDR corpus. Each feature of these four types is binary and its value is '1' or '0' ('1' denotes the presence of the corresponding feature, '0' its absence). The whole feature set shown in Table 2 is designed so as to cover the 210,000 sentences of EDR corpus.

[9]Terms of parts-of-speech tags and conjugation forms are borrowed from those of the Japanese morphological analysis system Chasen (Matsumoto et al., 1997).

### 3.4 Decision List Learning of Dependency Preference of Subordinate Clauses

First, in the modeling of the evidence, we consider every possible correlation (i.e., dependency) of the features of the subordinate clauses listed in section 3.3. Furthermore, since it is necessary to consider the features for both of the given two subordinate clauses, we consider all the possible combination of features of the anterior and posterior head vp chunks of the given two subordinate clauses. More specifically, let $Seg_1$ and $Seg_2$ be the head vp chunks of the given two subordinate clauses ($Seg_1$ is the anterior and $Seg_2$ is the posterior). Also let $\mathcal{F}_1$ and $\mathcal{F}_2$ be the sets of features which $Seg_1$ and $Seg_2$ have, respectively (i.e., the values of these features are '1'). We consider every possible subset $F_1$ and $F_2$ of $\mathcal{F}_1$ and $\mathcal{F}_2$, respectively, and then model the evidence of the decision list learning method as any possible pair $(F_1, F_2)$.[10]

Second, in the modeling of the decision, we distinguish the two cases of dependency relations described in section 3.1. We name the first case as the decision "modify", while the second as the decision "beyond".

### 3.5 Example

Figure 4 illustrates an example of transforming subordinate clauses into feature expression, and then obtaining training pairs of an evidence and a decision from a bracketed sentence. Figure 4 (a) shows an example sentence which contains two subordinate clauses $Clause_1$ and $Clause_2$, with chunking, bracketing, and dependency relations of chunks. Both of the head vp chunks $Seg_1$ and $Seg_2$ of $Clause_1$ and $Clause_2$ modify the sentence-final vp chunk. As shown in Figure 4 (b), the head vp chunks $Seg_1$ and $Seg_2$ have feature sets $\mathcal{F}_1$ and $\mathcal{F}_2$, respectively. Then, every possible subsets $F_1$ and $F_2$ of $\mathcal{F}_1$ and $\mathcal{F}_2$ are considered,[11] respectively, and training pairs of an evidence and a decision are collected as in Figure 4 (c). In this case, the value of the decision $D$ is "beyond", because $Seg_1$ modifies the sentence-final vp chunk, which follows $Seg_2$.

[10] Our formalization of the evidence of decision list learning has an advantage over the decision tree learning (Quinlan, 1993) approach to feature selection of dependency analysis (Haruno et al., 1998). In the feature selection procedure of the decision tree learning method, the utility of each feature is evaluated independently, and thus the utility of the combination of more than one features is not evaluated directly. On the other hand, in our formalization of the evidence of decision list learning, we consider every possible pair of the subsets $F_1$ and $F_2$, and thus the utility of the combination of more than one features is evaluated directly.

[11] Since the feature 'predicate-conjunctive-particle(chunk-final)' subsumes 'predicate-conjunctive-particle(chunk-final)-"*ga*"', they are not considered together as one evidence.

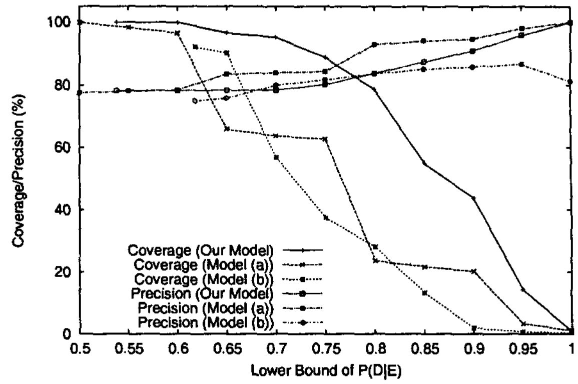



Figure 5: Precisions and Coverages of Deciding Dependency between Two Subordinate Clauses

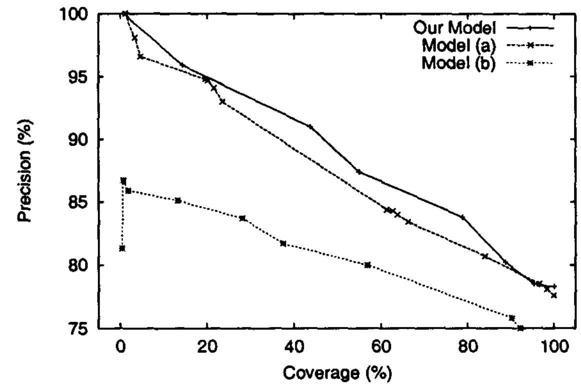



Figure 6: Correlation of Coverages and Precisions

## 4 Experiments and Evaluation

We divided the 210,000 sentences of the whole EDR bracketed Japanese corpus into 95% training sentences and 5% test sentences. Then, we extracted 162,443 pairs of subordinate clauses from the 199,500 training sentences, and learned a decision list for dependency preference of subordinate clauses from those pairs. The default decision in the decision list is $D =$"beyond", where the marginal probability $P(D =$ "beyond"$) = 0.5378$, i.e., the baseline precision of deciding dependency between two subordinate clauses is 53.78 %. We limit the frequency of each evidence-decision pair to be more than 9. The total number of obtained evidence-decision pairs is 7,812. We evaluate the learned decision list through several experiments.[12]

First, we apply the learned decision list to deciding dependency between two subordinate clauses of the 5% test sentences. We change the threshold of the probability $P(D \mid E)$[13] in

[12] Details of the experimental evaluation will be presented in Utsuro (2000).

[13] $P(D \mid E)$ can be used equivalently to the likelihood

the decision list and plot the trade-off between coverage and precision.[14] As shown in the plot of "Our Model" in Figure 5, the precision varies from 78% to 100% according to the changes of the threshold of the probability $P(D \mid E)$.

Next, we compare our model with the other two models: (a) the model learned by applying the decision tree learning method of Haruno et al. (1998) to our task of deciding dependency between two subordinate clauses, and (b) a decision list whose decisions are the following two cases, i.e., the case where dependency relation holds between the given two vp chunks or clauses, and the case where dependency relation does not hold. The model (b) corresponds to a model in which standard approaches to statistical dependency analysis (Collins, 1996; Fujio and Matsumoto, 1998; Haruno et al., 1998) are applied to our task of deciding dependency between two subordinate clauses. Their results are also in Figures 5 and 6. Figure 5 shows that "Our Model" outperforms the other two models in coverage. Figure 6 shows that our model outperforms both of the models (a) and (b) in coverage and precision.

Finally, we examine whether the estimated dependencies of subordinate clauses improve the precision of Fujio and Matsumoto (1998)'s statistical dependency analyzer.[15] Depending on the threshold of $P(D \mid E)$, we achieve 0.8~1.8% improvement in chunk level precision, and 1.6~4.7% improvement in sentence level.[16]

## 5 Conclusion

This paper proposed a statistical method for learning dependency preference of Japanese subordinate clauses, in which scope embedding preference of subordinate clauses is exploited. We evaluated the estimated dependencies of subordinate clauses through several experiments and showed that our model outperformed other related models.

ratio.

[14] Coverage: the rate of the pairs of subordinate clauses whose dependencies are decided by the decision list, against the total pairs of subordinate clauses, Precision: the rate of the pairs of subordinate clauses whose dependencies are *correctly* decided by the decision list, against those *covered* pairs of subordinate clauses.

[15] Fujio and Matsumoto (1998)'s lexicalized dependency analyzer is similar to that of Collins (1996), where various features were evaluated through performance test and an optimal feature set was manually selected.

[16] The upper bounds of the improvement in chunk level and sentence level precisions, which are estimated by providing Fujio and Matsumoto (1998)'s statistical dependency analyzer with correct dependencies of subordinate clauses extracted from the bracketing of the EDR corpus, are 5.1% and 15%, respectively.

## References

M. Collins. 1996. A new statistical parser based on bigram lexical dependencies. In *Proceedings of the 34th Annual Meeting of ACL*, pages 184–191.

EDR (Japan Electronic Dictionary Research Institute, Ltd.). 1995. *EDR Electronic Dictionary Technical Guide.*

J. Eisner. 1996. Three new probabilistic models for dependency parsing: An exploration. In *Proceedings of the 16th COLING*, pages 340–345.

M. Fujio and Y. Matsumoto. 1998. Japanese dependency structure analysis based on lexicalized statistics. In *Proceedings of the 3rd Conference on Empirical Methods in Natural Language Processing*, pages 88–96.

M. Haruno, S. Shirai, and Y. Oyama. 1998. Using decision trees to construct a practical parser. In *Proceedings of the 17th COLING and the 36th Annual Meeting of ACL*, pages 505–511.

J. Lafferty, D. Sleator, and D. Temperley. 1992. Grammatical trigrams: A probabilistic model of link grammar. In *Proceedings of the AAAI Fall Symposium: Probabilistic Approaches to Natural Language*, pages 89–97.

Y. Matsumoto, A. Kitauchi, T. Yamashita, O. Imaichi, and T. Imamura. 1997. Japanese morphological analyzer ChaSen 1.0 users manual. Information Science Technical Report NAIST-IS-TR97007, Nara Institute of Science and Technology. (in Japanese).

F. Minami. 1974. *Gendai Nihongo no Kouzou.* Taishuukan Shoten. (in Japanese).

J. R. Quinlan. 1993. *C4.5: Programs for Machine Learning.* Morgan Kaufmann.

S. Shirai, S. Ikehara, A. Yokoo, and J. Kimura. 1995. A new dependency analysis method based on semantically embedded sentence structures and its performance on Japanese subordinate clauses. *Transactions of Information Processing Society of Japan*, 36(10):2353–2361. (in Japanese).

T. Utsuro. 2000. Learning preference of dependency between Japanese subordinate clauses and its evaluation in parsing. In *Proceedings of the 2nd International Conference on Language Resources and Evaluation.* (to appear).

D. Yarowsky. 1994. Decision lists for lexical ambiguity resolution: Application to accent restoration in Spanish and French. In *Proceedings of the 32nd Annual Meeting of ACL*, pages 88–95.