Efficient Semantic Constraint for Speech Recognition

Yasuhisa Niimi* and Yutaka Kobayashi†

Abstract

In this paper we present a method for describing semantic constraint in SUSKIT-II, the speech understanding system we are developing. We integrate both syntactic and semantic constraints in a definite clause grammar. The syntax is expressed by a set of rewriting rules and the semantics by arguments of nonterminal symbols and auxiliary predicates on some arguments. These arguments, called semantic parameter, detail words semantically. An auxiliary predicate is a relation among semantic parameters to describe the simultaneous occurrence of semantic word classes. Moreover, using the relation, we introduce a concept hierarchy on the domain formed by a set of values which each semantic parameter can take. Since unnecessary resolution in the semantic level is avoided by means of this device, an amount of memory to store the linguistic structures of intermediate hypotheses could be reduced by 78.5 % in comparison with the case that the concept hierarchy was not used.

1 Introduction

In this paper we present a method for describing semantic constraint in a speech understanding system using the rule-based language model. By the rule-based language model we mean the model in which linguistic constraint is described by a set of deterministic rewriting rules. In SUSKIT-II, the speech understanding system we are developing, we integrate both syntactic and semantic constraints in a definite clause grammar. The definite clause grammar is the grammar in which each nonterminal symbol can have arguments and auxiliary predicates can be inserted between any two consecutive symbols in the right hand side of a rule. In SUSKIT-II the syntax is expressed by a set of rewriting rules and the semantics by arguments of nonterminal symbols and auxiliary predicates on some arguments. These arguments, called semantic parameter, play a role of the semantic marker and detail words semantically. An auxiliary predicate is a relation among semantic parameters to describe the possibility that semantic word classes occur simultaneously in a sentence.

Speech understanding systems generally preserve a number of partial sentences as intermediate hypotheses. In SUSKIT-II the intermediate hypotheses are preserved in the two

*Yasuhisa Niimi (新美 康永): Professor, Department of Electronics and Information Science, Faculty of Engineering and Design, Kyoto Institute of Technology

†Yutaka Kobayashi (小林 豊): Associate Professor, Department of Electronics and Information Science, Faculty of Engineering and Design, Kyoto Institute of Technology

(a) Kinkakuji no haikanryo wa gohyakuen desu ka.
(Is the entrance fee of Kinkakuji temple 100 yen ?)

(b) Hakubutsukan no kaikanjikoku wa juji yori hayai desu ka.
(Is the opening time of the museum earlier than 10 o'clock ?)

Figure 1: Example of acceptable sentences.

layers of tree structure. The one layer stores partial sentences as word strings, and the other stores their syntactic and semantic structures. We call the former word tree and the latter category tree. Thus multiple partial sentences with the same linguistic structure can share a node of the category tree. In this situation, the finer the semantic description of words is made to reduce the perplexity, the more decreases the number of the partial sentences that a linguistic structure can dominate. In other words, the more increases the number of nodes of the category tree.

To remedy this difficulty we introduce a concept hierarchy on the domain formed by a set of values which each semantic parameter can take. Since unnecessary resolution in the semantic level is avoided by means of this device, an amount of memory to store the linguistic structures of intermediate hypotheses can be reduced without increasing the perplexity. In the experiment which was conducted to test the proposed method, an amount of memory for the category tree could be reduced by 78.5 % in comparison with the case that the concept hierarchy was not used.

Section 2 gives a brief explanation of SUSKIT-II with emphases on the structure of the search space, and section 3 outlines the syntax of the task in SUSKIT-II and explains how to integrate the semantic constraint into the syntactic rules. Section 4 is devoted to the definition of the concept hierarchy on the semantic parameter and the procedure to organizing the concept hierarchy. Section 5 gives a brief result on the quantitative evaluation of the proposed method, and some discussions together with future work.

2 The Speech Recognition System

We suppose that a user of SUSKIT-II could issue oral questions about the contents of a relational database which contains information on sightseeing spots in Kyoto like temples, shrines, museums and hotels. Fig. 1 shows two examples of the questions. The relational database consists of tables like the one shown in Fig. 2. The name of each table is considered as a relation. A column contains values of the arguments of the relation, called attributes. A row corresponds to a record, an instance of the relation, which is a list of the values of the attributes.

Here we will give the belief explanation of the speech recognition system, SUSKIT-II [1],[2] with emphasis on the structure of the search space. Fig. 3 depicts the configuration of SUSKIT-II, which is composed of four components: acoustic processor, lexical matcher,

attribute name 1	attribute name 2	...
attribute value 11	attribute value 21	...
attribute value 12	attribute value 22	...
...	...	...

Figure 2: A relational Table.

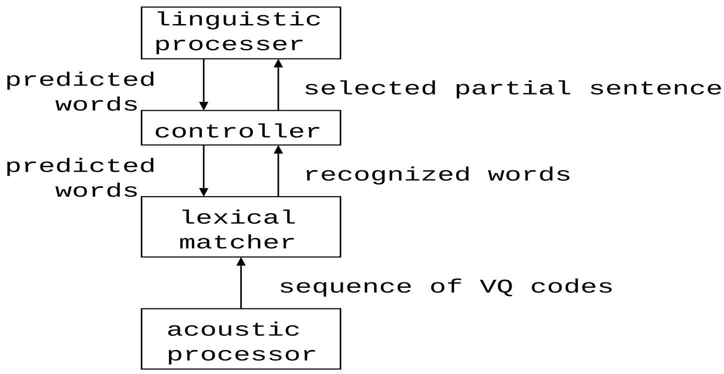


Figure 3: Configuration of SUSKIT-II.

linguistic processor, and controller. The acoustic processor calculates the LPC-cepstral coefficients , their time derivatives, and a pair of short-term energy and its time derivative for each 10 ms of speech, and then vector-quantizes them separately.

Given the templates of words hypothesized by the linguistic processor, each being a concatenation of phoneme-like hidden Marokv models, the lexical matcher verifies them against a portion of the acoustic data stream produced by the acoustic processor. It returns to the controller the words gaining scores higher than the threshold.

The linguistic processor makes top-down hypotheses, each consisting of a set of words capable of following a partial sentence selected by the controller. The partial sentence is a string of already recognized words covering the beginning portion of an input utterance.

The controller invokes other components while extending likely partial sentence hypotheses. It preserves partial sentences and their linguistic interpretations, both separately stored in the tree structure. They are referred to as 'word tree' and 'category tree', respectively.

The relation between the word tree and the category tree is shown in Fig. 4. A branch of the word tree represents a word, and a path (a sequence of branches) from the root node

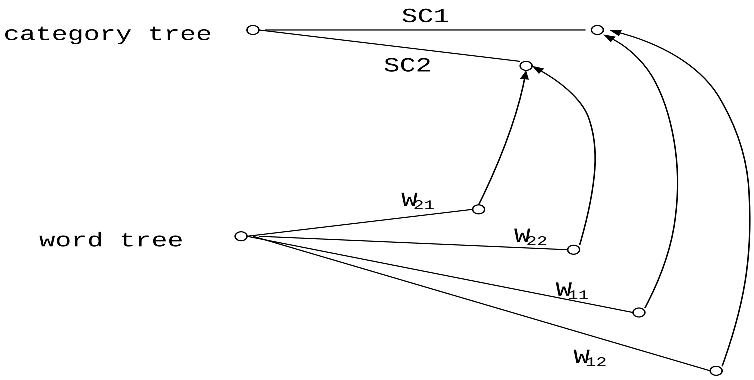


Figure 4: Memory STructure of Intermediate Hypotheses.

to a node represents a partial sentence. Leaf nodes, each corresponding to a partial sentence to be expanded, are ordered according to the matching score. The partial sentence with the best score will be expanded next. The expansion of a partial sentence means hypothesization the linguistic processor makes. This could be done by syntactic and semantic analyses of the partial sentence. For efficient hypothesization, however, linguistic interpretation of each partial sentence should be preserved and reused. The category tree is used for this purpose. Each node of this tree can be pointed from two or more nodes of the word tree. This means that the partial sentences corresponding to these nodes have the same linguistic interpretation.

There is an alternative for preserving syntactic and semantic descriptions of partial sentences. In this approach the linguistic information is stored directly in each node of the word tree. The latter is simple in structure, but needs more memory than the method we adopted. Since memory was expensive when we began to develop SUSKIT-II, we adopted the method which uses two layers of the tree.

3 Syntax and Semantics

3.1 Task syntax and semantics

Fig. 5 illustrates syntactic rules for query sentences by a definite clause grammar (DCG). The terminal symbols of this grammar are underlined and the nonterminal ones are not. The symbols in the parentheses are semantic parameters. The terms enclosed by the brackets expresses the predicates for the semantic constraint. The vocabulary contains about 250 words. The test set perplexity is 8.3 for the current task.

s →	cp(C), fn(wa) ,ppc(C,q) | cp(C), fn(wo),v(find)
	| tp(T), fn(wa) ,ppt(T,q) | tp(T), fn(wo),v(find)
tp(T) →	tcst(T) | cat(T,C),cp(C), fn(hap),tn(C,T)
	| {cat(T,C)},cp(C),{verbfr(C,F,T,V)}, fn(F),v(V,n)
cp(C) →	cst(C) | cn(C),uch ,ppc(C,rt), mono
	| ppc(C,rt), cn(C)
ppc(C,q) →	{cat(T,C)}, qn(T) ,{verbf(T,F,C,V)}, fn(F) , v(V,q)
ppt(T,S) →	cfp(T,F), adj(T,F,S) | dp(T),aux(T,S)
sp(T,C) →	tn(T,C),fn(subj)
cfp(T,F) →	tp(T),{compf(T,F)}, fn(F)
dp(T) →	tp(T) | cfp(T,F), an(T,F)

Figure 5: A part of syntactic rules of the sightseeing task.

3.2 Syntactic structures and categorization of words

The nonterminal symbols describing the above grammar reflect the specific features of the database queries. Besides the starting symbols s, we introduced seven nonterminals.

cp: noun phrases which specify the search key in the relational tables.

tp: noun phrases which specify the attributes in the relational tables.

ppc, ppt: predicative phrases containing a verb which terminates a sentence or modifies a noun phrase. ppc modifies cp and ppt does tp.

sp: noun phrases which specify the main topic in a ppc phrase.

cfp: phrases which represent an object of comparison.

dp: noun phrases which specify case fillers in ppc or ppt phrases.

We categorized the words in the vocabulary into fourteen task specific categories. Especially the nouns were subcategorized according to the concepts related to the database. The words other than the noun were classified according to the standard school gram mar. The categorization of the noun is described below.

cn: : names of relational tables in the database.

tn: names of attributes in a relational table.

cst: proper nouns appearing in the column of the name attribute.

tcst: nouns appearing in the columns other than the name attribute.

qn: interrogative nouns.

3.3 Semantic parameters and case structures

In order to examine whether or not a pair of words can appear in a syntactic structure, we associate parameters with the syntactic rules. These parameters, called semantic parameter, play roles of the semantic marker. The two types of semantic constraints between two noun phrases are used in the present system. Letting A, B, C and D be noun phrases, we explain these constrains as follows.

(1) For the compound noun phrase, "A no B (B of A)," we impose a constraint that B is an attribute name of a record specified by the noun phrase A. Both specified items must exist in the same relational table.

(2) In the sentence, "C wa D desu ka (Is C D ?)", which is frequently used in the present task, we imply that C and D refer to an attribute name and its value, respectively. Both C and D must be relevant to the same attribute of the database, for example, "cost" and "500 yen". An example is given by the sentence (a) in Fig. 1.

We use semantic parameters C and T for describing these constraints. The syntactic categories 'cn' and 'cst' are augmented by the parameter C whose value is one of the names of relational tables. The syntactic categories 'tn' and 'qn' are augmented by the parameter T whose value is one of higher concepts of the attribute names. The predicate cat(T,C) in Fig. 5 expresses the constraint that the value of T must be an attribute of the relational table specified by C.

We describe the semantic (or co-occurrence) relation between an adjective or a verb and other phrases by the semantic parameter and the case grammar. In the present task the case frame of both an adjective and a verb has two case slots. One of the two slot fillers plays a role of the syntactic subject. The other forms the predicative phrase together with a postposition and an adjective or a verb.

The sentence (b) in Fig. 1 shows an example of how adjectives are used in the present task. They are used only to compare two items (the opening time and 10 o'clock in the example) which are semantically described by the parameter T. Thus the adjective can be characterized by the semantic parameter T. The relation between the adjective and the postposition can be tabulated. The predicate compf(T,F) in Fig. 5 expresses such a relation, where the parameter F denotes the class of postpositions.

The two slot fillers of a verb are described by semantic parameters T and C. The predicates verbf(T,F,C,V) and verbfr(C,F,T,V) in Fig. 5 express a relation among these two semantic parameters T and C, the class of verbs (denoted by V), and the postposition which can be attached to the slot filler other than the subject.

4 The Concept Hierarchy

4.1 Memory Reduction by the Concept Hierarchy

In order to reduce the perplexity of a task, it is necessary to use as much linguistic knowledge as possible. We have made an attempt at reducing the perplexity by detailing semantic description of words, because the fine classification of words decreases the average number of words capable of following partial sentences syntactically and semantically, by which the perplexity is measured.

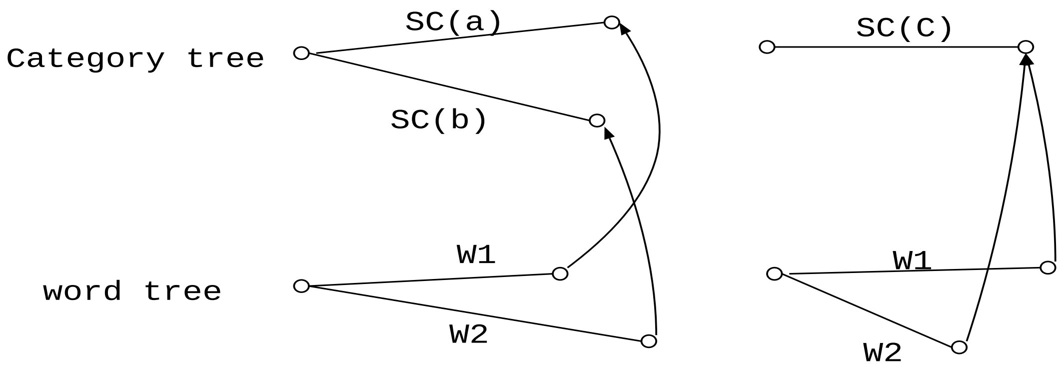
In the method we used to preserve partial sentences, the more the semantic description of words is detailed, the more decreases the number of the nodes of the word tree that share a node of the category tree. Thus the number of nodes of the category tree increases. There are two reasons for this. The first is that if two words belonging to the same semantic class in the coarse semantic description are separated into two different classes in the detailed description, two nodes of the category tree are necessary for these two words. The second is related to the possibility for two words to simultaneously occur in a sentence. Consider the case in which co-occurrence of two words is described by a semantic parameter. For example, in a noun phrase composed of an adjective and a noun , a semantic parameter x can describe the possibility for the two to consecutively occur as noun_phrase(x) $\rightarrow$ adjective(x),noun(x). If the adjective can modify two kinds of nouns, described as noun(a) and noun(b) (a and b denote two different values of the semantic parameter x), it must have two different descriptions adjective(a) and adjective(b). For example, the adjective 'onaji ('same' in English) can modify the nouns meaning person, time, place, and cost. This means two nodes of the category tree are necessary for this adjective.

In this paper we propose a method for detailing semantic description of words without increasing the number of nodes of the category tree. For this purpose we structure a concept hierarchy on the domain formed by a set of all values a semantic parameter can take. The basic idea is shown in Fig. 6 and Fig. 7. Assume that two words W1 and W2 in a syntactic category SC belong to two different semantic classes SC(a) and SC(b) respectively. Then we need two nodes SC(a) and SC(b) of the category tree as shown in Fig. 6-(a). If we define an upper concept 'c' of 'a' and 'b' capable of co-occurring with 'a' and 'b', two nodes can share a node SC(c) as shown in Fig. 6-(b). Fig. 7 shows another example, in which a word W with two semantic descriptions SC(a) and SC(b) requires two corresponding nodes of the category tree. The introduction of the upper concept 'c' with the property as stated above can make a node SC(c) represent two different semantic descriptions.

4.2 Automatic Organization of the Concept Hierarchy

We define upper concepts on a domain, based on the four semantic constraints, cat(T,C), compf(T,F), verbf(T,F,C,V), and verbfr(C,F,T,V). The procedure can be stated as follows, in which the explanation is given for the domain of the semantic parameter T as an example.

(1) Partition of the domain



(a) W1 and W2 have their own node in the category tree.

(b) W1 and W2 share a node in the category tree.

Figure 6: The case a syntactic category SC contains two words 'W1'and 'W2' which belong to different semantic classes

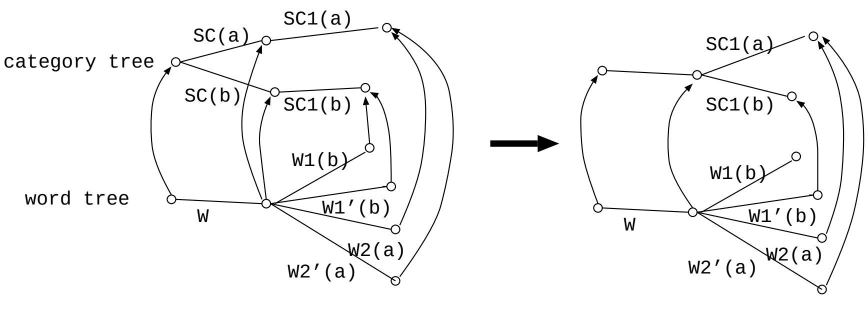


Figure 7: The case a word 'w' belongs to two semantic classes.

Let D(T) denote the domain of T and D(C) denote the domain of C. For any t D(T) and c D(C), the truth value of the predicate cat(t,c) is given. The 'true' means t and c can occur simultaneously in a phrase, and the 'false' means they cannot.

(1-1) The domains D(T) and D(C) are divided into disjoint subsets $\{T_i\}$ and $\{C_k\}$ respectively satisfying the following conditions.

(i) $D(T) = \bigcup_i T_i \quad T_i \cap T_j = \phi (i \neq j),$

$D(C) = \bigcup_k C_k \quad C_k \cap C_l = \phi (k \neq l),$

where ϕ is the empty set.

(ii) Either cat(t,c)=true for all (t,c)'s such that t $\in T_i$ and c $\in C_j$, or cat(t,c)=false for such (t,c)'s.

We call the collection of subsets $\{T_i\}$ thus obtained a partition of D(T). It follows from the above conditions (i) and (ii) that each subset in the partition of D(T) can be treated as a unit which there is no necessity for further dividing under the semantic constraint cat(T,C).

(1-2) The domain D(T) is divided in the same way based on other semantic constraints relating to the semantic parameter T. Thus we have several partitions of the domain D(T), denoted by $P^{(i)}(T)$. For example, we have four partitions of D(T), each being based on one of the four semantic constraints.

(2) Integration of partitions

Those partitions are integrated one by one to build the finest partition. During the integration process, upper concepts are introduced based on the inclusion relation between subsets of two partitions to be integrated.

(2-1) Let $A = \{Ai\}$ be $P^{(1)}(T)$.

(2-2) The following steps (a) to (d) are repeated with changing I from 2 to N.

(a) Let $B = \{B_j | j = 1..b\}$ be $P^{(I)}(T)$ and X be $\{\phi\}$.

(b) The following procedure is repeated with changing j from 1 to b. Let $A_{jk}(k = 1..m_j)$ be all subsets of A that have common elements with B_j , that is, $X_{jk} = A_{jk} \cap B_j \neq \phi(k = 1..m)$ and $B_j = \bigcup_k X_{jk}$. These intersections X_{jk} 's are appended to X as its members, and B_j is defined as an upper concept of X_{jk} 's if m_j is greater than one.

(c) The step (b), in which A and B are exchanged in their role, is repeated. In this step no members are added to X, but some A_i 's are defined as upper concepts.

(d) Let A be initialized with X and go to the step (a) with adding one to I.

(2-3) The final X with upper concepts is the integrated partition.

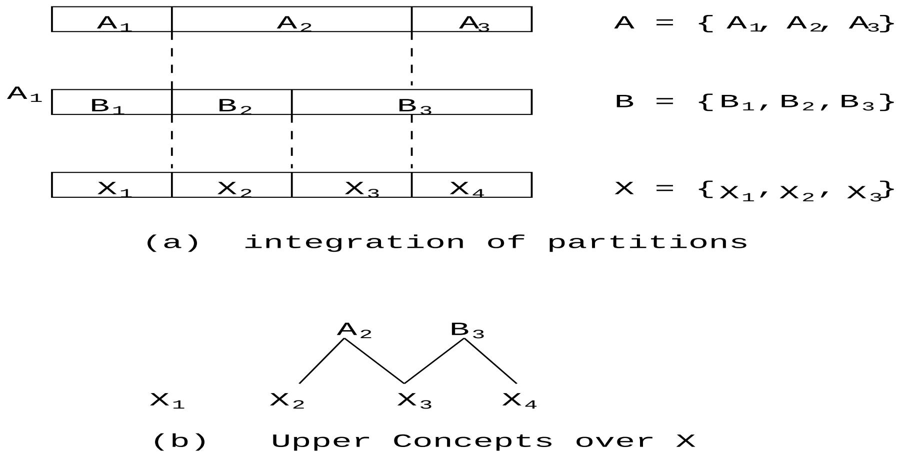


Figure 8: Integration of partitions and Upper concepts.

Some explanation should be made for the step (b) in the above procedure. Assume that $P^{(1)}(T)$ and $P^{(2)}(T)$ be derived from the semantic constraints C_1 and C_2 respectively. The step states that a subset B_j of $P^{(2)}(T)$ must be divided into X_{jk} 's under the semantic constraint C_1 if m_j is greater than one. The subset B_j is, however, defined as an upper concept over X_{jk} 's, because it is the finest subset of values of T under the semantic constraint C_2 . Fig.8 explains the steps for the integration of partitions. Two partitions $A = \{A_1, A_2, A_3\}$ and $B = \{B_1, B_2, B_3\}$ of a domain are illustrated in Fig. 8-(a). For these partitions the integrated one is $X = \{X_1, X_2, X_3, X_4\}$ where $X_1 = A_1 = B_1$, $X_2 = B_2 = A_2 \cap B_2, X_3 = A_2 \cap B_3$, and $X_4 = A_3 = A_3 \cap B_3$. Two upper concepts A_2 and B_3 are defined as in Fig. 8-(b).

5 Evaluation and Discussions

For the quantitative evaluation of the proposed method for the semantic representation of intermediate hypotheses, we organized two knowledge sources; the one (called KS-1) which uses the concept hierarchy on the semantic parameter, and the other (called KS-2) which does not use it. Both KS's have the same linguistic power, that is, their perplexities are the same. We made SUSKIT-II recognize 53 oral questions using those KS's, and counted the number of the nodes of the category tree which had been expanded during the recognition in both cases. The number does not mean the amount of memory used for the category tree, because there are left the nodes of the category tree which have not been expanded. However, the ration of the number of the expanded nodes to the number of the unexpanded ones might be considered almost the same for both KS's. Moreover, the number of the expanded nodes is given a precise measure of the time spent for the linguistic processing. For these reasons we counted the number of the expanded nodes of the category tree. These numbers were 3922 for KS-1 and 18262 for KS-2. This means that the introduction of the concept hierarchy reduces an amount of memory and time spent for the linguistic processing by 78.5 %.

In this paper we have proposed a method for the semantic processing in a speech recognition system. The method premises that the semantic constraint is given as a relation among semantic parameters. The relation is used to divide into disjoint subsets the domain formed by a set of all values a semantic parameter can take. Using the concept hierarchy on the domain which is organized based on the partition of the domain, we can reduce remarkably memory space and processing time spent for intermediate hypotheses produced in the process of speech recognition. While the quantitative evaluation of it was made for the small task (the vocabulary size is about 250), we expect similar effect for larger tasks. Although the semantic constraint tested was strongly dependent on the task, the method can be applied to any semantic constraint as long as it is expressed as a relation among semantic parameters.

However, it would become more difficult to manually define the semantic constraint for a larger task. In computational linguistic research several attempts have been made at classifying words semantically and constructing a thesaurus on basis of a large amount of text corpus or the dictionary [3]-[7]. It is future work to adapt these studies to the generation of linguistic knowledge which can be used in speech recognition, that is, which reduces the perplexity.

References

[1] Y.Kobayashi, M.Omote, H.Endo, and Y.Niimi, "SUSKIT-II — A speech understanding system based on robust phone spotting," IEICE Trans. Vol.E-74, No.7, pp.1863-1869 (1991).

[2] Y.Niimi, and Y.Kobayashi, "An information retrieval system with a speech interface," Proc. of ICSLP-92, pp.1407-1410 (1992).

[3] P.F.Brown, V.J.D.Pietra, P.V.deSouza, J.C.Lai, and R.L.Mercer,"Class-based n-gram modeling of natural language," Computational Linguistics, Vol.18, No.4, pp.467-479 (1992).

[4] D.Hindle,"Noun classification from predicate argument structure," Proc. of the 28th Annual Meeting of ACL, pp.268-275 (1990).

[5] F.Pereira, N.Tishby, and L.Lee, "Distributional clustering of English words," Proc of the 30th Annual Meeting of ACL (1992).

[6] F.R.Framis, "An experiment on learning appropriate selectional restriction from a parsed corpus," Proc of COLING-94, pp.769-774 (1994).

[7] H.Tsurumaru, H.Maeda, K.Yamamoto, T.Hitaka, and S.Yoshida, "A study on thesaurus construction by using Japanese language dictionary," Tech. Report of IEICE, NCL93-58 (1993) (in Japanese).