自然言語処理 128−12
(1998. 11. 5)

SD式意味モデルを用いた意味差の尺度の計算

吉原 将太† 脇山 正博†† 河口 英二†

†九州工業大学 ††北九州工業高等専門学校

意味構造記述式モデル (Semantic-structure Description Form Semantics Model) は，著者らによって開発された自然言語の意味を定量的に分析するための枠組みである．このモデルに従った意味記述をSD式と呼ぶ．SD式モデルの特徴は，与えられた2つの概念が意味的にどの程度近いかということを定量的に扱えることである．

著者らは，自然言語世界における概念の意味量を計算するために，詳述関係を定義している．本稿では，このモデルを使った意味差の尺度の計算について述べる．意味量計算の概要，および与えられた概念から最近共通先祖を発見する方法を示す．

Computation of Semantic Difference Measure using SD-Form Semantics Model

Shouta Yoshihara† Masahiro Wakiyama†† Eiji Kawaguchi†

†Kyushu Institute of Technology

††Kitakyushu National College of Technology

The SD-Form Semantics Model, developed by the authors, is a framework to analyze the meaning of natural language in a quantitative way. It is equipped with a formal language named SD-Form which describes the semantic structure of each language expression.

We define an elaboration relation in order to calculate the semantic metric of two concepts in natural language. The objective of the present paper is to show a semantic difference measure under this model. We describe the semantic information score computation scheme, as well as the nearest common ancestor detection process from a given concept pair.

1. Introduction

There are many models to describe concept relation and sentence meaning of natural language. Among all, frame models and network models are very common to us [1,2]. Those who are interested in logical formulation advocate the first-order predicate logic [3,4]. However, nobody has succeeded in building a practical system based on those models in natural language processing.

Previously, the authors proposed a new semantics model using SD-Forms (Semantic structure Description Forms) as a meaning description language [5,6,7,8]. We call it SD-Form Semantics Model. The syntax of each SD-Form is defined by a context-free grammar (SDG). The SD-Form model provides formalization of intelligent operations in a knowledge system such as concept recognition, understanding, inference, etc. The authors have already reported several feasibility studies on the applications [9,10,11]. The most important point of the model is that it is equipped with scheme for semantic metric computation in terms of semantic difference measure.

This scheme is associated with an elaboration relation between two concepts. When a concept (D_2) is a detailed idea of some other concept (D_1), we say that they have an elaboration relation such that D_2 is an elaborated concept from D_1. "An elaboration score" measures the amount of elaboration. The elaboration relation depends not only on the syntactic similarity of two SD-Forms but also depends on the knowledge data that is available in the system. The semantic difference measure between two concepts is defined by finding their common meaning, which is termed the nearest common ancestor.

The idea of the elaboration relation and the nearest common ancestor are the fundamentals of our model. The objective of the present paper is to discuss the way we have implemented the computation algorithm of the semantic difference measure under given knowledge.

We briefly review in Section 2 how to describe the semantic structure of concepts using SD-Forms. In Section 3, we give a general idea of concept elaboration and present how we can compute the nearest common ancestor of the two given concepts. In Section 4, we will explain the experimental system named "SDENV-3". It is a program package written by Prolog. Finally in Section 5, we give our conclusions and show the problems for our future work.

2. Concept description by SD-Forms

The SD-Form Semantics Model defines the SD-Form as a meaning description language. It assumes that when human communicate with others by natural language the idea is always definite, but the language expression can be ambiguous. Actually we definitely know what we want to say, but there are many options how we express it. This brings ambiguity to the sentence meaning. Our assertion is that even if we do not know a language to express our brain, its conceptual structure in our mind should be describable by some means. The SD-Form semantics model gives one of such means.

2.1 The syntax of the SD-Form

As we have described in the Introduction, We have a "meaning description language" named "SD-Form". An SD-Form is a string of symbols. It is either "atom", "term", or "list" of Prolog programming. Each SD-Form is

formally generated by a context-free grammar (SDG), which has no ambiguity. Terminal symbols of the SDG include concept labels, a modifier, prescriptors, connectors, functional items, and parentheses [5]. The followings are examples of concept descriptions.

(1) Concept label

Concept labels are primitive symbols in an SD-Form. They do not have any structural information by themselves. We often take advantage of English or Japanese words for such labels. Concept variables, such as X, Y, Z are also concept labels.

<Ex.2-1>

DOG, TARO, JAPAN, BUY, SOME, ANY, 5, DOLLAR(100), X, Y, Z

(2) Modified SD-Forms

When some concept has more semantic information than a single label, we make a more detailed description by adding a " / " followed by a modifying single SD-Form, or a set of SD-Forms connected by " *para* "'s. They are called "modified SD-Form".

<Ex.2-2>

GIRL/PRETTY (A pretty girl)

STONE/(HEAVY)*para*(BIG)

(A big heavy stone)

(3) Prescribed SD-Forms

Prescriptors prescribe semantic role of the concepts. We have five prescriptors, namely, "*nega*", "*pass*", "*assu*", "*only*", "*fcus*".

<Ex.2-3>

nega(GIVE) (do not give) ···*Negation*

pass(HIT) (be hit) ···*Passive form*

assu([*s*(TOM),*v*(COME)]) (if Tom comes)

···*Assumption*

(4) Connected SD-Forms

Connectors are connective operators to combine two SD-Forms to make a new composite concept. We have more than 30 different connectors such as "*incl*", "*equa*", "*andx*" etc.

<Ex.2-4>

(ANIMAL)*incl*(MONKEY)

(Animal include Monkey)

(BOB)*equa*(FATHER/MARY)

(Bob equals to Mary's father)

(LIGHT)*andx*(SMALL)

(light and small)

(5) Statement SD-Forms

A statement SD-Form is a list of SD-Forms with statement functional items. They are;

s: Subject item, *v*: Predicate item,

i: Indirect object item, *o*: Object item,

c: Complement item, *b*: Agent item

We have 9 types of statement SD-Forms.

<Ex2-5>

- [*s*(JIM),*v*(DRINK/PAST),*o*(BEER)] (Jim drank beer.)
- [*s*(KEN),*v*(GIVE),*i*(ERI),*o*(FLOWER)] (Ken gives Eri a flower.)

(6) Emotion SD-Forms

"Greetings", "Calling and Responses", "Emotional utterances" and many other non-statement expressions convey very important information in verbal communications. We have three types of functional items for emotion SD-Forms.

a: Attention item, *r*: Reply item,

e: Exclamation item

<Ex.2-6>

- [*a*(MARK)] (Hi, Mark.) ···*Calling*
- [*r*(NEGATIVE)] (No.) ···*Reply*
- [*e*(SURPRISE)] (Wow!) ···*Exclamation*

2.2 Knowledge description in an intelligent machine

We install all the knowledge pieces in a machine by using SD-Forms [5]. They are limited SD-Forms such as the followings.

(1) Prescribed SD-Form

nega(D)

(2) Connected SD-Form

(*assu*(D_1))*caus*(D_2), (*assu*(D_1))*indu*(D_2),

$(D_1)incl(D_2)$, $(D)csof([D_{11},D_{22},\cdots,D_{1n}])$,
$(D_1)equa(D_2)$, $(D_1)defi(D_2)$

(3) Statement SD-Form

$[s(D_1),v(D_2)]$, $[s(D_1),v(D_2),c(D_3)]$,
$[s(D_1),v(D_2),o(D_3)]$, $[s(D_1),v(D_2),i(D_3),o(D_4)]$,
$[s(D_1),v(D_2),o(D_3),c(D_4)]$

3. Elaboration relation between two concepts

The idea of elaboration relation between two concepts is the most important in SD-Form Semantics Model. We first introduce a notion of quantitative semantic information carried by an SD-Form.

3.1 Semantic information of an SD-Form

Each SD-Form (D) carries a certain amount of semantic information. This amount depends on the syntactic structure of D. We designate it by *si*(D) and call it the semantic information score of D. Although the score assignment details are left open to each model user, we have some general ideas about it.

(1) *si*(D) should be the accumulation of the scores of partial SD-Forms in D. Therefore, a simple SD-Form has a small score, while a complicated one a large score.

(2) Each SD-Form symbol should be equipped with a primary score to initiate score computation.

(3) The primary score of a concept label should have the largest value among other symbols, because a concept label simulates a "words", and words are key information in natural language. We should give a flat score to all of them, except for a variable, because they only designate "concept symbols". Further modification will be possible.

(4) The absolute value of *si*(D) does not mean much. We are more concerned with the relative score of each concept.

The followings show the primary scores in our experimental system (SDENV-3) we employed. The unit of the score is termed "semit (semantic information unit)".

A. A variable label has 1 semit.
B. Each simple concept label has 10 semits.
C. The modifier " / " has 1 semit.
D. Each prescriptor has 2 semits.
E. Each connector has 1 semit.
F. Each functional item has 1 semit.
G. " [ ] " has 1 semit.
H. " ( ) " and " , " has 0 semits.

These scores are determined in consideration of the basic ideas ((1)~(4)) presented above, as well as the requirements of abductive inference in a system. However, we are not claiming these score as our essential idea for the SD-Form Semantics Model, rather they are all test score in our experiment.

<Ex.3-1>

- si(ORANGE/BIG) = 21 (a big orange)
- si((BOY)$plus$(GIRL/PRETTY)) = 32 (boy and pretty girl)
- si([s(TOM),v(DRIVE),o(CAR/NEW)]) = 45 (Tom drives a new car)

3.2 Elaboration in the SD-Form Semantic Model

An elaboration relation in SD-Form Semantics Model is a generalized idea of the traditional "IS-A", "PART-OF" or "IF-THEN" relations. Elaboration from one D_1 to anther D_2 means that D_2 is a more specific or detailed expression of D_1. It has two types, one is syntactic, and the other is knowledge based. The elaboration scores quantitatively measure the degree of elaboration. It is taken as a measure of uncertainty when we abductively infer D_2 out of true D_1.

If D_2 is an elaborated form of D1, We say D_1 and D_2 have an "elaboration relation". We denote by,

$elab(D_1,D_2) = n$, or $elab(D_1,D_2,n)$.
(n:elaboration score, $0 \leqq n < \infty$)

where, n is the elaboration score given by;

$n = min\{ elab_{synt}(D_1,D_2), elab_{know}(D_1,D_2) \}$

We will spell out $elab_{synt}(D_1,D_2)$ and $elab_{know}(D_1,D_2)$ in the following sections.

3.2.1 Syntactic elaboration relation

When D_1 and D_2 are related in one of the following cases, D_2 is called "syntactically elaborated" from D_1 by the score n.

$elab_{synt}(D_1,D_2) = n$, or $elab_{synt}(D_1,D_2,n)$.

(1) D_2 is generated from D_1 by SDG rule. In this case the elaboration score is

$elab_{synt}(D_1,D_2) = si(D_2) - si(D_1)$.

(2) D_1 is a label and D_2 is of the form D1/D.

$elab_{synt}(D_1,D_2) = si(D) + 1$.

The formal definition of the syntactic elaboration relation is given in other paper [5].

<Ex.3-2>

- $elab_{synt}$(BOOK, BOOK/PRECIOUS) = 11
- $elab_{synt}$([s(HUMAN), v(EAT)], [s(HUMAN), v(EAT), o(FOOD)]) = 11

3.2.2 Knowledge-based elaboration

Knowledge-based elaboration relation depends on knowledge, which links D_1 and D_2 semantically. There are two types of knowledge-based elaboration;

1. Specific-knowledge-based elaboration
2. General-knowledge-based elaboration

Specific-knowledge-based elaboration is based on individual knowledge available in the system.

A. $(D_i)equa(D_j)$ → $elab_{know}(D_i,D_j) = 0$

B. $(D_i)defi(D_j)$ → $elab_{know}(D_i,D_j) = 0$

C. $(D_i)csof([D_{j1},\cdots,D_{jk},\cdots,D_{jn}])$ → $elab_{know}(D_i,D_{jk}) = 2$

D. $(assu(D_j))caus(D_i)$ → $elab_{know}(D_i,D_j) = 2$

E. $(D_i)incl(D_j)$ → $elab_{know}(D_i,D_j) = 3$

F. $(D_i)ptof(D_j)$ → $elab_{know}(D_i,D_j) = 3$

G. $(D_i)kdof(D_j)$ → $elab_{know}(D_i,D_j) = 3$

<Ex.3-3>

- (HUMAN)$csof$([MAN, WOMAN]) → $elab_{know}$(HUMAN, MAN) = 2
- (FRUIT)$incl$(APPLE) → $elab_{know}$(FRUIT, APPLE) = 3

While, the general-knowledge-based elaboration relation is associated with our general knowledge about SD-Form quantifier usage. The respective score setting was made experimentally in relation to the specific knowledge-based score setting.

$elab_{know}(D_i/X, D_i/SOME) = 1$
$elab_{know}(D_i/SOME, D_j) = 1$
$elab_{know}(D_i/SOME, D_k) = 1$
$elab_{know}(D_i/SOME, D_i/MOST) = 3$
$elab_{know}(D_i/SOME, D_i/ANY) = 4$
$elab_{know}(D_i/SOME, D_i/D) = 2$
$elab_{know}(D_i/MOST, D_i/ANY) = 1$
$elab_{know}(D_i/D, D_i/ANY) = 2$
$elab_{know}(D_j/D, D_i/ANY) = 2$
$elab_{know}(D_k/D, D_i/ANY) = 2$
$elab_{know}(D_j, D_i/ANY) = 3$
$elab_{know}(D_k, D_i/ANY) = 3$

<Ex.3-4>

Let the System knowledge be

(HUMAN)$csof$([MAN,WOMAN]),
(KID)$kdof$(BOY)

In this case,

$elab_{know}$(HUMAN/SOME, WOMAN) = 1
$elab_{know}$(KID, BOY/ANY) = 3

3.3 Multi-step recursive elaboration

A set of elaboration relations can be concatenated into a multi-step elaboration. In the case of an m-step elaboration, we designate it by $elab_m(D_i,D_2)$. Because an elaboration is either syntactic or knowledge-based, and a concatenation of syntactic elaborations is always reduced to a single step, the number of combinations of syntactic (denoted by (S)) and knowledge-based (denoted by (K)) relations in an m-step elaboration is less than 2^m. For m =1,2,3, all

the m-step elaboration types are the following. We call each combination pattern a "path type".

1 step: D_1-(K)-D_2, D_1-(S)-D_2

2 step: D_1-(K)-(K)-D_2,
D_1-(K)-(S)-D_2, D_1-(S)-(K)-D_2

3 step:
D_1-(K)-(K)-(K)-D_2, D_1-(K)-(K)-(S)-D_2,
D_1-(K)-(S)-(K)-D_2, D_1-(S)-(K)-(K)-D_2,
D_1-(S)-(K)-(S)-D_2

<Ex.3-5>

If we have fact data, such that,

(KYUSHU)*ptof*(JAPAN/WEST)

Then KYUSHU should be an elaborated concept of JAPAN, because

$elab_{synt}$(JAPAN, JAPAN/WEST),

$elab_{know}$(JAPAN/WEST, KYUSHU),

relation hold true. And the elaboration relation should be transitive. This example is a two-step elaboration;

$elab_2$(JAPAN, KYUSHU)
= $elab_{synt}$(JAPAN, JAPAN/WEST)
+ $elab_{know}$(JAPAN/WEST, KYUSHU)
= 11 + 3 = 14

3.4 The nearest common ancestor

Let D, D_1, D_2 be three concepts in SD-From, which satisfy;

$elab(D, D_1) = n_1$ and $elab(D, D_2) = n_2$.

We call such D a common ancestor of D_1 and D_2. D can be either $D=D_1$ or $D=D_2$, but not equal to both. A simple example of such D is a variable "X". It is a common ancestor of any two non-variable SD-Forms.

One of the common ancestors D_0 which is the nearest to D_1 and D_2 is called the nearest common ancestor ("*ncoa*" in short) of D_1 and D_2. We describe this relation as;

$ncoa(D_1, D_0, D_2, n_1, n_0, n_2)$

where,

$n_o = n_1 + n_2 = elab(D_0, D_1) + elab(D_0, D_2)$
$= min_D\{ elab(D, D_1) + elab(D, D_2)\}$. ···①

When we do not care about the score, we simply describe it as;

$ncoa(D_1, D_0, D_2)$.

The "semantic difference measure" between two concepts is given by the following.

A. The case $D_1 = D$ and $D_2 = nega(D)$ are two concepts where D is an arbitrary SD-Form;

$diff(D_1, D_2) = \infty$

B. Otherwise;

$diff(D_1, D_2) = n_0$

where n_0 is computed by ①. We call this measure a semantic difference score.

<Ex.3-6>

Let the system facts be;

(CAT)*incl*(TAMA) (Cat includes Tama.)

(Animal)*incl*([CAT, HUMAN, MOUSE, COW])
(Animal includes cat, human, mouse and cow.)

(HUMAN)*incl*(SAM) (Human includes SAM.)

(BEEF)*kdof*(COW) (Beef is a kind of cow.)

(*assu*((X)*kdof*(HUMAN)))
caus((FRIEND/X)*kdof*(HUMAN))
(If X is a human, then any X's friend is a human.)

and two given concepts be;

D_1=[*s*(TAMA), *v*(EAT/PAST), *o*(MOUSE)]
(Tama ate a mouse.)

D_2=[*s*(FRIEND/SAM), *v*(EAT), *o*(BEEF/FRESH)]
(Sam's friend eats some fresh beef.)

In this case, the nearest common ancestor is the following.

D_0=[*s*(ANIMAL/SOME), *v*(EAT),
o(ANIMAL/SOME)]
(Some animal eat some animal.)

4. Experiment

4.1 Experiment System (SDENV-3) for SD-Form Semantics Model

We have developed an experimental system of the SD-Form Semantics Model. It is titled as SDENV-3, a third version of our prototype system. It is a Prolog program, which works under Windows.

The important functions of the SDENV-3 are as follows.

A. Appending new labels, facts, and rules.

B. SD-Form syntax checking and *si* computation.

C. $elab_{synt}$, $elab_{know}$ and $elab_m$ score computation for a concepts pair.

D. *ncoa* and *diff* computation.

4.2 Experiment for *ncoa* detection

Let D_1 and D_2 be two "given concepts".

D_1=[*s*(JIRO), *v*(DRINK/EVERYDAY), *o*(BEER)]
(Jiro drinks beer everyday.)

D_2=[*s*(EMI), *v*(BUY/PAST/YESTERDAY), *o*(WINE/WHITE)]
(Emi bought a white wine yesterday.)

The "Facts" in the system (Individual and General) are as follows.

F_1=(PERSON)*incl*([FATHER,JAPANESE])
(Person includes father and Japanese.)

F_2=(ALCOHOL)*incl*([BEER,WINE])
(Alcohol includes beer and wine.)

F_3=(JAPANESE)*incl*(EMI)
(Japanese includes Emi.)

F_4=(JIRO)*equa*(FATHER/KEN)
(Jiro is a Ken's father.)

F_5=[*s*(JIRO), *v*(DRINK/EVERYDAY), *o*(BEER)]

F_6=[*s*(EMI), *v*(BUY/PAST/YESTERDAY), *o*(WINE/WHITE)]

General "Rules" in the system are

R_1=(*assu*([*s*(X), *v*(DRINK/EVERYDAY), *o*(Y)]))
caus([*s*(X), *v*(LIKE), *o*(Y)]),
(If X drinks Y everyday, then X likes Y.)

R_2=(*assu*([*s*(X), *v*(BUY), *o*(Y)]))
caus([*s*(X), *v*(LIKE), *o*(Y)]).
(If X buys Y, then X likes Y.)

Under this circumstance the two following concepts (induced concepts) work as system knowledge (specific rules) after instantiating general rules with facts.

R_1'=(*assu*([*s*(JIRO), *v*(DRINK), *o*(BEER)]))
caus([*s*(JIRO), *v*(LIKE), *o*(BEER)])

R_2'=(*assu*([*s*(EMI), *v*(BUY), *o*(WINE)]))
caus([*s*(EMI), *v*(LIKE), *o*(WINE)])

By using this system knowledge, the *ncoa* of D_1 and D_2 is detected by the algorithm described in Section 4.1. All the elaboration relations are illustrated in Fig.4. The numbers are elaboration scores. Knowledge symbols ($F_1,\cdots,F_7$, R_1, R_2, R_1', $R2'$) are also attached to each (*K*). The overall elaboration form D_0 to D_1 takes 2 steps, while the one from D_0 to D_2 takes 3 steps. Therefore, the semantic difference measure in this case is computed as;

$diff(D_1, D_2) = elab_2(D_0, D_1) + elab_3(D_0, D_2)$
$= (((1+11+0)+1)+2)+(((1+3)+1)+2+(22+11))$
$= 55$

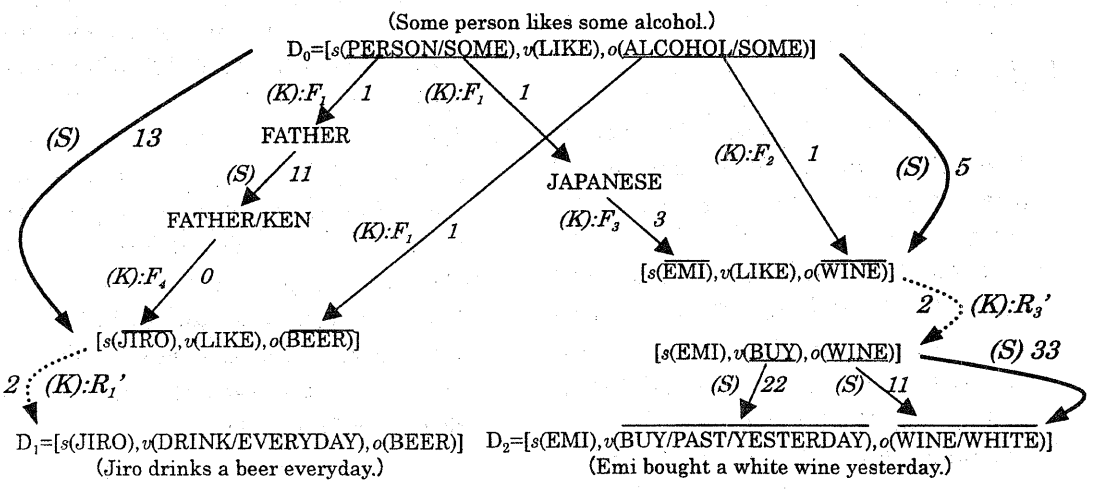

Fig.4 An example of the *diff*(D_1, D_2) computation

5. Conclusions

The elaboration relation is the most fundamental idea in the SD-Form Semantics Model. It is either a syntactic or knowledge-based one. The degree of elaboration from one concept (D_1) to another (D_2), i.e., the elaboration score, provides the semantic discrepancy between them. It is a generalized idea of the traditional "IS-A", "PART-OF" and "IF-THEN" relations.

In this paper, we described the detail of the elaboration relation with the emphasis of elaboration score computation scheme. Also, we defined the nearest common ancestor of two concepts, and showed how to detect it in the system.

The algorithm of the semantic difference computation will open many ways to meaning processing of natural language, such as in recognition, understanding and learning.

Some of the problems we should work on in the near future are the following.

A. Testing the model in a more realistic world, even if it is small.

B. An abstracting algorithm of story data written by SD-Forms.

C. Advancement of SDENV-3 development more intensively.

Reference

[1] Quillian,M.R. : "Semantic memory, in Semantic Information Processing", edited by Minsky, M.L, MTP Press, pp.227-270, 1968.

[2] Rada,R., Mili,H., Bicknell,E. and Blettner,M. : "Development and application of a metric on semantic nets", IEEE Transaction on System, Man, and Cybernetics, 19,pp.17-30, 1989.

[3] Montague,Richard : "Proper Treatment of Quantification in Ordinary English", in Formal philosophy, edited by Thomason, R.H., Yale University Press, pp.247-270. 1974.

[4] Clocksin,W.F. and Mellish,C.S. : "Programming in prolog, Third Revised and Extended Edition", Springer-Verlag Berlin Heidelberg, New York, 1987.

[5] Kawaguchi,E., Wakiyama,M. and Nozaki,K. : "A Semantic Structure Description Model of General Concepts in a Natural Language World", Proc. of PRICAI, pp.298-303,1990.

[6] Kawaguchi,E., Kamata,S. and Wakiyama,M. : "Elaboration Relation and the Nearest Common Ancestor of a Concept Pair in the SD-Form Semantics Model", Proc. of 2nd PRICAI, pp.426-432,1992.

[7] Kawaguchi,E. Kamata,S. and Wakiyama,M. : "The Semantic Metric Computation Scheme in the SD-Form Semantics Model", PROC. of 3rd PRICAI, pp.623-629, 1994.

[8] Wakiyama,M., Shao,G., Nozaki,K. and Kawaguchi,E. : "The Toward Generalization of the Semantic Metric in the SD-Form Semantics Model", Proc., of 4th PRICAI'96 Poster, pp.61-68, 1996.

[9] Wakiyama,M., Shao,G., Nozaki,K. and Kawaguchi,E. : "Anaphora Processing of Story Data by the SD-Form Semantics Model Approach", Proc.PRICAI,pp.1169-1175, 1992.

[10] Wakiyama,M., Yoshihara,S. and Kawaguchi,E. : "A Prototype of Japanese Sentence Generation System form SD-Formed Meaning Data", Proc. of the PACLING'97, pp.333-344, 1997

[11] Wakiyama,M. and Kawaguchi,E. : "Retrieval of Multimedia Data for Natural Language in the Network", Proc. of 4th PRICAI'98 Poster, 1998