

# Diagnostic Processing of Japanese for Computer-Assisted Second Language Learning

Jun'ichi Kakegawa, Hisayuki Kanda, Eitaro Fujioka, Makoto Itami, Kohji Itoh

Department of Applied Electronics,

Science University of Tokyo

2641 Yamazaki, Noda-shi, Chiba-ken 278-8510, JAPAN

{kakegawa,kanda,eitaro76,itami,itoh}@itlb.te.noda.sut.ac.jp

## Abstract

As an application of NLP to computer-assisted language learning(CALL) , we propose a diagnostic processing of Japanese being able to detect errors and inappropriateness of sentences composed by the students in the given situation and the context of the exercise texts. Using LTAG(Lexicalized Tree Adjoining Grammar) formalism, we have implemented a prototype of such a diagnostic parser as a component of a CALL system being developed.

## 1 Introduction

In the recent classroom of second language learning, *communicative approach*(H.G. Widowson, 1977) is promoted in which it matters for the students to become aware of the language use, i.e. the functionality of language usage and it's dependence on the situations and the contexts of communication. In order to achieve the objective according to "constructivistic" point of view of learning (T.M.Duffy *et al.*, 1991), the students are encouraged to produce sentences by themselves in various situations and contexts and guided to recognize by themselves the erroneous or inappropriate functions of their misused expressions.

We have already proposed a *Computer-Assisted Language Learning(CALL)* system (N.Kato *et al.*, 1997) which provides the students with sample texts promoting their re-

flection on the errors and inappropriateness, detected by a diagnostic parser, of the sentences composed by the students filling the blanks set up in the given contexts and situations. In this paper we report on prototyping the diagnostic parser implemented using LTAG formalism as a component of the system.

LTAG(Lexicalized Tree Adjoining Grammar) is a lexicalized grammatical formalism (XTAG Research Group, 1995). For ease of diagnosing the erroneous sentences composed by the students, lexicalized type of grammars seemed most suitable. Comparing HPSG(Head-driven Phrase Structure Grammar) (C.Pollard *et al.*, 1994) and LTAG, the well-known two (almost-)lexicalized grammars, LTAG looked more simple and especially convenient for sentence generation necessary in diagnosis. LTAG systematically associates an elementary tree structure with a lexical anchor and the structure is embedded in the corresponding lexical item. Associated with each of the external nodes of the embedded tree structure are feature structures such as inflection, case information, head symbol, semantic constraints as well as a difference list for surface expressions. These features have their origin in the anchored lexical item. The feature information can, moreover, include the knowledge of situated language use. Appearance of the features at the external nodes of the lexical items greatly facilitates generation of local phrases which is indispensable in diagnostic parsing. These are the reason why we employed LTAG.

Preference of unification to all-procedural handling excluded the so-called "dependency

grammar” (M.Nagao, 1996).

## 2 LTAG of Japanese

### 2.1 The Characteristic of Japanese

Japanese phrases are classified in the first place into two categories: Yougen phrase(YP) and Taigen phrase(TP). A YP or TP has a Yougen or a Taigen, respectively, as it’s head word. Yougen along with Taigen as categories belong to the category of semantically self-contained (called autonomous) words. The words, e.g. verbs, adjectives, belonging to Yougen have inflections, whereas the words. e.g. nouns, pronouns, demonstratives, belonging to Taigen have no inflection. A YP or TP consists of a head word and its sibling phrases on it’s left semantically modifying the head word. And such a phrase in its turn can semantically modify an autonomous word by way of attaching a connective to it’s right, forming a phrase, or inflecting the head word of the modifier.

In general, a sentence is constructed by attaching to a phrase a few (or void of) functional words expressing the attitude of the locutor to the propositional part of the phrase ( modality ) and intention of the locution affecting the listener ( illocutionary-act marking ).

### 2.2 Elementary Tree

Fig.1 shows *Elementary Trees* of LTAG we defined for Japanese.

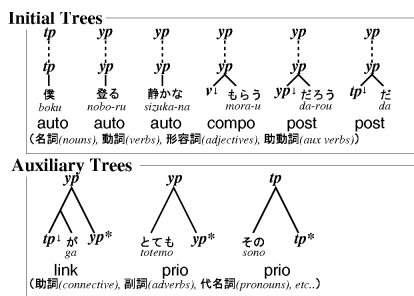


Figure 1: Example of Elementary Trees

Each node is expressed by a *predicate* formalism, in general, as following,

```
yp(<Head Variable>,<Inflection>,<Used Case Information>,<Unused Case Information>,<Semantic Constraint>,<Semantic Relationship>).
```

```
tp(<Head Variable>,<Semantic Constraint>,<Semantic Relationship>).
```

For example, “登る” is a self-contained (autonomous) word and its lexical item, comprising an initial tree, is expressed by,

```
auto(“登る”,
yp(X,終止,
[[Z,[“人(Z)”,が],[Y,[“場所(Y)”,“に”]],SlashCase,
[“行為(X)”,[“登る”(X,Z,Y)|SemXMod]],
yp(X,終止,[[Z,[“人(Z)”,が],[Y,[“場所(Y)”,“に”]],
[“行為(X)”,[“登る”(X,Z,Y)]]]).
```

Note that tense, aspect, polite expressions, “Ren-you て (te)” are dealt with as inflections just as in the classes teaching Japanese as Second Language. The lexical items are classified into several categories such as **auto**, **link**, **prio**, **post**, **compo**, according to the embedded tree structures.

### 2.3 Tree Operation

In LTAG, 2 tree operations are defined(See Fig. 2). A node of a tree is said to be substituted by another tree if the root node of the latter is successfully unified with the node. A tree is said to be adjoined with another tree if it is successfully inserted into the latter by unifying the root node and the foot node(marked \*) of the former, respectively, with the separated nodes of the latter, all with a same syntactic category.

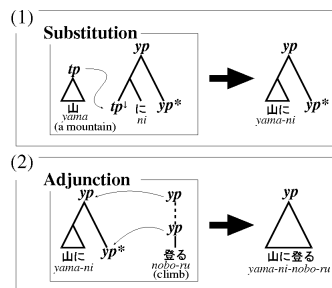


Figure 2: Examples of Substitution and Adjunction

In Japanese, a *Yougen* requires as ad-

joined modifiers Taigen phrases with connectives(e.g. Fig. 2 (1)) corresponding to the mandatory “cases” ( e.g. Fig.2 (2) ), and it also require have those corresponding to the optional “cases”.

The default order of the case phrases may be changed for the purpose of stressing or avoiding unintended modification. The change can be dealt with by way of permutation in unification.

Another type of phrase to modify the Yougen is YP plus one of the connectives denoting cause, reason-why, condition etc.(e.g. Fig.3 (4)).

A Yougen may be modified by a YP (Yougen Phrase) with its head Yougen inflection in Ren-you form without any connective(e.g. Fig.3 (3)).

A Taigen is mostly modified by a YP (Yougen Phrase) with its head Yougen inflected in Rentai form with no connective(e.g. Fig.3 (2)).

For ease and uniformity of processing, especially in the diagnostic parser, null connectives  $\lambda$ -Ren-you and  $\lambda$ -Rentai are introduced when a YP modifies Yougen and Taigen, respectively, by way of inflection(e.g. Fig.3 (3), (2) ).

The other type of phrase to modify the Taigen is TP plus connective “ の (*no*)” denoting proprietary, kinship or whole-part relationship(e.g. Fig.3 (1)).

## 2.4 Dealing with Situation - Dependent Expression

By incorporating into the feature structure an additional item expressing situational constraints, the parser has the capability of diagnosing usage of situation-dependent Japanese expressions such as giving and receiving benefits as well as demonstratives. As for demonstratives, e.g. “この本 (*kono-hon*)”, “その本 (*sono-hon*)”, “あの本 (*ano-hon*)” indicates a book located either in the territory of the locuter, the listener, or outside the both, respectively.

In the case of expression for giving and receiving benefits, for example as shown in Ta-

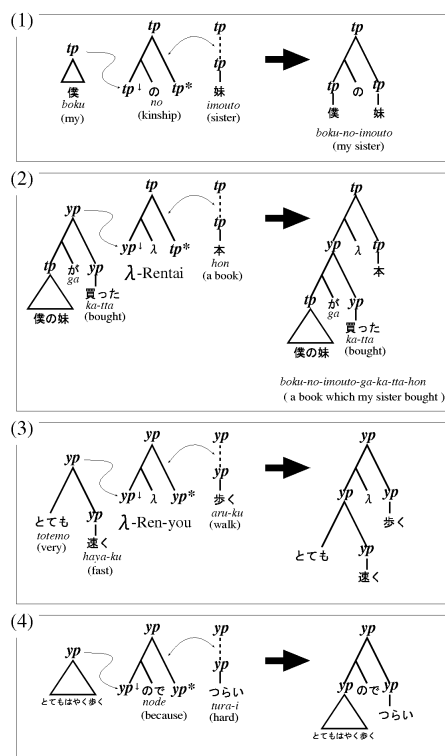


Figure 3: Examples of Tree Structure

ble 1, the empathy relational constraints are embedded in each of the lexical items for the underlined word along with the case information for “*が* (*ga*)”, “*に* (*ni*)”

Though the indicated three expressions have the same propositional function of expressing giving-benefit whose giver is  $x$  and givee is  $y$ , “camera” is placed on the side of  $x$ ,  $y$ ,  $y$  with “angles” towards  $y$ ,  $x$ ,  $x$  respectively. It is seen that the camera angle determines the requirement to the empathy relations(S.Kuno, 1989).

Suppose the situation  $E(X|Z) < E(Y|Z)$  is given, where  $X$ ,  $Y$ ,  $Z$  stand for “the nurse”, “the locutor’s son”, “the locutor”, respectively, for instance, the parser can diagnose the following.

**English :**

“ The nurse(:X) reads the book to my son(:Y). ”  
: I(:Z) am the locutor.

**Japanese : incorrect**

“ 保母さん(:X)が私の息子(:Y)に本を読んであげる。 ”  
(*hobo-san ga watashi no musuko ni hon wo yonde*)

Table 1: Situational Constrains in Lexicon

| Expressions   | Case Information | Empathy constraint |
|---|------------------|--------------------|
| $x$ が $y$ に $\sim$ して <u>あげる</u> ( $x$ ga $y$ ni $\sim$ shite-ageru)  | $x$ が, $y$ に     | $E(x z) > E(y z)$  |
| $x$ が $y$ に $\sim$ して <u>くれる</u> ( $x$ ga $y$ ni $\sim$ shite-kureru) | $x$ が, $y$ に     | $E(x z) < E(y z)$  |
| $y$ が $x$ に $\sim$ して <u>もらう</u> ( $y$ ga $x$ ni $\sim$ shite-morau)  | $y$ が, $x$ に     | $E(y z) > E(x z)$  |

locutor  $z$  :  $x$  give\_benefit  $\sim$  to  $y$

ageru . )

**Japanese : correct**

“ 私の息子(:Y)が保母さん(:X)に本を読んでもらおう。 ”

(*watashi no musuko ga hobo-san ni hon wo yonde-morau .*)

## 2.5 Composite Verbs

The above-mentioned expressions for giving and receiving, e.g. “読んでもらおう” *yonde-morau*, is an example of “composite verbs” in Japanese.

Many composite verbs can be produced with a considerable number of auxiliary verbs preceded by different main verbs.

Because of the modification of the sense and the case control due to the auxiliary component, as illustrated in the case information column in Table1, we are forced to generate the composite tree (See Fig.4), carrying out modification of the meaning and the case control, before adjoining of modifiers to the composite verb takes place.

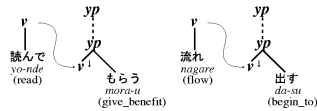


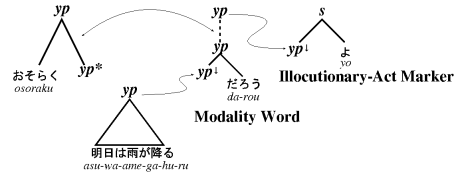
Figure 4: Examples of Composite Verb

## 2.6 Modality Words and Illocutionary - Act Markers

In Japanese, “modality words” are functional words expressing the attitude of the locutor towards the propositional part of the utterance, “illocutionary-act markers” demands

answer from the listener or expresses other intention of the locution affecting the listener.

Some combinations of certain adverbs and a “modality word” co-occur in the position interposing that part of the proposition in which the locutor has concern. The example shown in Fig.5, “だろう” (*darou*) is a modality word expressing locutor’s supposition, and “おそらく” (*osoraku*) expresses the extent of his confidence on the supposition. The lexical item for the latter includes the demand for the modality semantics of the locutor’s supposition.



**English :** It will probably rain tomorrow, I’m sure.

**Japanese :** 明日は、おそらく 雨が降る だろう よ。

(*ashita wa , osoraku ame ga huru darou yo .*)

Figure 5: Modality Word and Illocutionary-Act Marker

## 2.7 Connective “wa”

In Japanese, TP plus connective “は” (*wa*) is frequently used. It is said that there are two kinds of usage of connective “は” ; the one introduces the theme of the sentence, the other discriminatorily presents one of the cases of the head Yougen as shown, respectively, in the following cases.

**usage 1**

**English :** Me, I climbed that mountain.

**Japanese :** 僕はあの山に登った。  
(*boku wa ano yama ni nobo-tta.*)

**usage 2**

**English :** (e.g.) As for me, I'll have a dish of eel.

**Japanese :** 僕はウナギだ。  
(*boku wa unagi da.*)

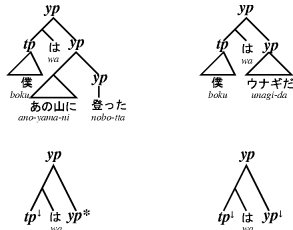


Figure 6: Example of the Usage of “wa”

In distinguishing between usage1. and usage2., we focus on the head Yougen of YP. If it has any unfilled-case, and the semantic constraint of the Taigen before connective “は” corresponds to that of one of the unfilled-cases, then our processor regards “は” as discriminatory.

Otherwise, “は” is considered as introducing the theme of the sentence.

**2.8 Use of a Stack in Parsing**

For implementing a parser for Japanese, a stack memory can be conveniently employed. ‡ In processing the sentence from left to right, the candidate modifier phrases are kept in a stack memory until a possible Yougen or Taigen word appears and inspected if they can modify the word. The tree-structured features of the candidate modifier phrases popped up one by one from the stack are tried to be unified with those of the word, and the features of the phrases as far as the tree adjoining unification succeeds are integrated with the features of the modified word, to make a *Saturated Initial Tree*(SIT). The rest of the phrases of the stack are left there

to be tested on the next Yougen or Taigen word which will appear later on. Any ordering of modifiers is syntactically permitted except when an undesired modification takes place.

‡ If a connective is found by reading one word ahead, the thus-far made SIT substitutes the left external node of the tree of the connective to make a *Saturated Auxiliary Tree*(SAT) provided unification succeeds(e.g. Fig.7). If the read ahead is a modality word, its yp node is substituted by the yp root of the SIT, and after interposing modality modifiers having been processed, the resulting phrase is considered SIT anew and the procedure goes to ‡. If the read ahead is an illocutionary-act marker or the ending sentence symbol, and the inflection of SIT is appropriate, parsing terminates. Otherwise either of the  $\lambda$ -Renyou or  $\lambda$ -Rentai connectives is attached depending on the inflection of the head of the SIT to make a SAT ( See Fig. 3 and Fig. 7). In either cases as well as the case with a non-null connective, the SAT is pushed into the stack and the procedure recurs to ‡.

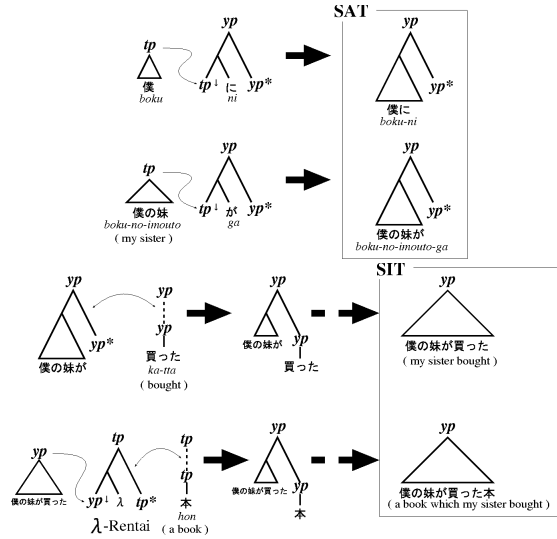


Figure 7: Example of SAT and SIT

**3 Generation**

We describe here our algorithm for generating a sentence when the semantic relation-

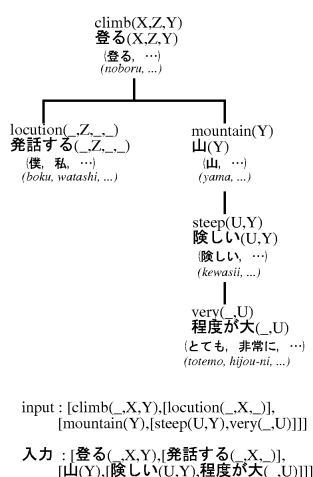


Figure 8: Example of a Semantic Relationship and Trees

ship, for example as in Fig8, is given. The generation process progresses as illustrated in Fig9.

The main stream of our generation algorithm follows.

At first, from the lexical database, an autonomous word is fetched, whose semantic relationship term is unifiable with the root of the given semantic relationship. Letting the root and terminal node of the word be the first and the second arguments, respectively, **generate2** is called

- If the first argument can be unified with the second argument, generation is terminated. Otherwise, the process, carrying over the second argument, searches for a **prio** or **link** word whose root node can be unified with the first argument.
- If a **prio** word is found, letting its right ( foot ) node be the first argument and retaining the second argument, **generate2** is called.
- If a **link** word is found, an autonomous word is searched for whose root node can be unified with the left ( substitution ) node of the **link** word. Letting the word's root and the terminal node be the first argument and the second argument,

respectively, **generate2** is called. Letting the right ( foot ) node of the link word be the first argument and retaining the second argument, **generate2** is called.

In the following, searching of the autonomous word and handing their 2 nodes off to **generate2** are dealt with by **generate1** predicates.

```
generate1(Node):-
    auto(W,Node,Terminal),
    generate2(Node,Terminal).
```

```
generate2(Node1,Node2):-
    unify(Node1,Node2).
```

```
generate2(Root,Terminal):-
    prio(W,Root,Right),
    generate2(Right,Terminal).
```

```
generate2(Root,Terminal):-
    link(W,Root,Left,Right),
    generate1(Left),
    generate2(Right,Terminal).
```

In the case of generation including modality words, illocutionary-act markers or composite verbs, the algorithm needs a little more complicated procedures.

## 4 Case and Semantic Processing in Parsing and Generation

In parsing and generation, case and semantic processing occurs by unification without any procedural programming.

The initial tree structure of the lexical item of an autonomous word consists of a root node and a terminal node.

Especially in the YP initial tree, the root node has a filled used-case slot and a variable unused-case slot as well as a variable semantic slot whose head part is filled. The terminal node has the null used-case slot and the filled unused-case slot as well as the semantic slot consisting only of the head predicate.



Figure 9: Example of Generation

In parsing, following the process as illustrated in Fig. 9 bottom up, when the foot YP node of a YP SAT ( e.g. とても 険しい 山 に ) is unified with the terminal node of a Yougen autonomous word ( e.g. 登る ), the case data, if any, ( e.g. [Y, [ 場所 (Y)], に ] ) corresponding to the SAT is moved from the unused-case slot to the used-case slot in the SAT root node. The semantic data from the SAT is integrated with that of the word and transferred to the SAT root. The foot YP node of another YP SAT if any, ( e.g. 僕 が ) is unified with the said root node the corresponding case data, if any, ( e.g. [Z, [ 人 (Z)], が ] ) is further moved from the unused-case slot to the used-case slot.

The semantic data from the new SAT is joined with that in the previous SAT root in the root of the new SAT.

Likewise proceeding, finally, by unifying the concatenated SAT with the root of the original autonomous word ( e.g. 登る ), there remains in the unused case slot those case datas with no corresponding SAT which may be explained by omitted SATs or the slash case whose entity will be found in the Taigen word to be modified by the thus-constructed modifying YP.

The whole semantic data from the SATs is integrated in the root node of the original autonomous word.

The process of adjoining TP SATs ( e.g. とても 険しい ) to modify a Taigen autonomous word ( e.g. 山 ) is similar to that for YP SATs to a Yougen word, except that no case data processing occurs.

In generation, following the process as illustrated in Fig. 9 top-down, when the whole given semantic relationship is unified into the semantic slot of a Yougen autonomous word ( e.g. 「 登る 」 ), and if e.g. a link word ( e.g. 「 が 」 ) is found with its root unifiable with the root of the Yougen initial tree, the semantic expression is divided into two parts thanks to the case data ( e.g. [Z, [ 人 (Z)], が ] ), the one part ( e.g. [ 発話する ( -, Z, -, -) ] ) is transferred to the right node, and the other part ( e.g. [ 登る (X, Z, Y), [ 山 (Y), [ 険しい (U, Y), 程度が大 ( -, U) ] ] ) transferred to the left ( foot ) node. From the case data of the used-case slot of the original yougen ( e.g. 「 登る 」 ), the case data corresponding to the link word ( e.g. 「 が 」 ) is moved from the used-case slot to the unused-case slot in the left ( foot ) node.

That part of semantics transferred to the right node is processed to find the corresponding surface expression ( e.g. 「 僕 」 ) by constructing an SIT. The other part of semantics sent to the left ( foot ) node along with the remaining used-case slot ( e.g. [[Y, [ 場所 (Y)], に ]] ) are made use of for finding a link word ( e.g. 「 に 」 ) whose root node is unifiable with the said left ( foot ) node.

The semantics sent to the new link root node is divided into two parts; the one part ( e.g. [ 山 (Y), [ 険しい (U,Y), 程度が大 (-,U)] ] sent to the right node to form SIT and construct the corresponding surface expression ( e.g. 「 とても 険しい 山 」 ), the other part ( e.g. [ 登る (X,Z,Y)] ) sent to the left ( foot ) node.

Likewise proceeding, when all the used-case data is transferred into the unused-case slot in the foot node, it may be unified with the terminal node of the original yougen ( e.g. 「 登る 」 ), terminating the generation.

## 5 Mechanism of Semantic Diagnostic Processing

### 5.1 Postulation

In our CALL system, the students are asked to fill in the blanks for composition in the given situation and context, using words from a given list. Therefore no morphological analysis is needed. In diagnosing the students' sentence, we assume that the following data is available for constraining processing.

- Semantic elements and their relationships, which should be expressed by the sentence with which the students are asked to fill the blanks.
- The list of words, to be used in the composition, corresponding to the semantic elements.

Fig.10 is an example of relationships of semantic elements represented by a tree structure. Modifying elements are placed as the children of the parent, the modified elements. The list of the words to be used for expressing an element is linked to the element.

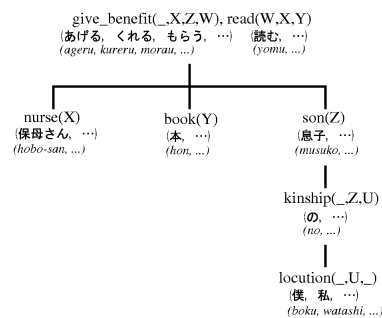


Figure 10: Example of relationships of semantic elements

### 5.2 Principle of Semantic Diagnosis

After an SIT has been constructed, the diagnostic parser consults the lexicon with the succeeding word. If it is a connective, the parser tries substitution operation with SIT and, if successful, appends it to the SIT to form the temporary SAT. In case the parser fails to append the connective to the SIT, only the surface expression of the connective along with the SIT is recorded in the provisional SAT. Suppose the succeeding word was not a connective. If it was a Taigen or Yougen and the SIT is yp and its inflections is Rentai or Ren-you, respectively, then  $\lambda$ -Rentai or  $\lambda$ -Ren-you is appended to the SIT to form an SAT, even though the inflection might be incorrect. If the inflection of the SIT is inconsistent with the succeeding word or the SIT is tp, as no reasonable interpretation is possible, “**Pending Connective**”  $\mu$  is appended to the SIT to make an SAT. In all of the above-mentioned cases, the obtained SAT is pushed into the stack. When the parser encounters a Yougen word[ $\#$ ] or a Taigen word, it pops up one SAT after another from the stack and examines, locally generating surface expressions, if it conforms with one of the semantic children to the parent corresponding to the target Yougen/Taigen word. If it does, the parser adjoins the SAT to the word, after, if necessary, having corrected **wrong/missing connective** or **wrong inflection** of the SAT, thus making an SIT, including error correction messages if any. If the popped SAT does



not conform with any of the semantic children, it is pushed into a temporary stack, recording the SAT as a **false modifier** if SAT can be falsely adjoined to the Yougen/Taigen word. In case of SAT accompanying  $\mu$ , the parser, consulting the semantic relationship tree data, generating a related phrase, either replaces  $\mu$  with a suitable **missing connective** and/or corrects the **wrong inflection** if necessary. When an SAT is popped up which conforms with one of the semantic children, the SATs held in temporary stack at that instance, if any, should have been **obstacles** for the popped up SAT to **modify** the target word. And they are marked “✖”. After all the SATs in the main stack have been examined, the SATs recorded in the temporary stack are returned into the main stack. And then the SAT constructed as explained in the above is pushed into the main stack. If, later on, the SATs marked “✖” are found to modify a target word, conforming to the semantic relationship, they are commented as causing **modification crossover**. Finally, if the semantic relationship requires modality expression(s) and/or illocutionary-act marker(s), the thus-far-made Yougen SIT is (recursively if necessary) substituted into the **yp** node of the expression(s) and, at the same time, corresponding modifiers of the expression(s) are looked for in the main stack to be popped making an SIT.

If at [#], the found Yougen word is a part of a composite verb the semantic relationship requires, the rest is looked for, supplemented if lacking, the case information is modified if necessary, and the same procedures follow as described after [#].

## 6 Example of Diagnosis

For example, supposing the student had input the sentence shown in Fig.11, the parser could detect the errors by using the semantic relationship aforementioned in Fig.10 and the relation of the degrees of empathy in the given situation.

The detected errors are listed in the following.

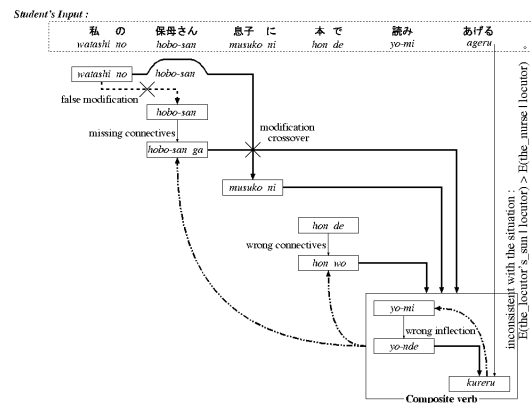


Figure 11: Example of Result of Diagnosis

### false modification :

Inappropriate placing “私の”(watashi no), causing the phrase to modify “保母さん”(hobo-san).

### missing connective :

Missing connective “が”(ga) which “保母さん”(hobo-san) must have for the phrase to be adjoined to “読んでくれる”(yo-nde kureru).

### obstacle for modification :

“保母さん”(hobo-san) is in the place of obstacle for “私の”(watashi no) to modify “息子”(musuko).

### wrong inflection :

“読み”(yo-mi) has to be replaced by “読んで”(yo-nde) for the verb to form a composite verb together with auxiliary verb “くれる”(kureru) expressing **giving benefit**.

### wrong connective :

Wrong connective “で”(de) has to be replaced by “を”(wo) which “本”(hon) must have for the phrase to be adjoined to “読んでくれる”(yo-nde kureru).

### modification crossover :

The sentence has a modification crossover between “私の息子”(watashi no musuko) and “保母さんが読んでくれる”(hobo-san ga yo-nde kureru).

### inappropriate situational expression :

Use of “あげる” (*ageru*) in the given situation designates empathy relation

$$E(\textit{nurse}|\textit{locutor}) \\ > E(\textit{the\_locutor's\_son}|\textit{locutor})$$

which contradict with the given empathy relation. It requires less number of corrections for “あげる” to be replaced by “くれる” (*kureru*) for conforming with the relation and retaining “息子に” (*musuko ni*) than to be replaced by “もらう” (*morau*).

## 7 Conclusions

We proposed a diagnostic processing of Japanese and described its procedures in detail. The parser makes use of LTAG formalism introducing several additional data structure such as SIT, SAT, null/pending connectives.

The diagnosis we reported here is local in principle. Referring to the given relationship of semantic elements, the error is detected and corrected locally. The correction messages are generated and recorded locally in SITs. The undesired modifications in the student sentence, however, can be detected and commented on. Our CALL system, based on the detected errors and inappropriateness, provides the students with sample texts which will enable the students to correct their sentence by themselves.

The tasks to be achieved are:

1. to establish ontology of semantic relationship description,
2. efficient methodology for preparing the lexical items comprising semantic constraints,
3. to communicate semantic contexts and situations to the students through assisting reading the texts by way of bidirectionally linking the text words with an electronic dictionary,
4. to deal with anaphora.

## Acknowledgment

The authors are grateful to Prof. Jun-ichi Tsujii, University of Tokyo, for discussing and providing information on LTAG as well as the status quo of natural language processing.

The work reported in this paper was partially supported by the Grant-in-Aid for Scientific Research 09680303, Ministry of Education.

## References

- The XTAG Research Group ( 1995 ) : “ A Lexicalized Tree Adjoining Grammar for English ”, University of Pennsylvania, IRCS Report 95-03, March 1995.
- Owen Rambow and Aravind K. Joshi ( 1994 ) : “ A Processing Model for Free Word Order Languages ”, In *Perspectives on Sentence Processing*, C.Clifton, Jr.,L.Frazier and K.Rayner, editors. Lawrence Erlbaum Associates.
- Carl Pollard, Ivan A. Sag ( 1994 ) : “ Head-Driven Phrase Structure Grammar ”, The University of Chicago Press.
- M.Nagao ( 1996 ) : “ Natural Language Processing ”,Iwanami-Shoten.
- V. M. Holland, J. D. Kaplan, M. R. Sams ( 1995 ) : “ Intelligent Language Tutors – Theory Shaping Technology – ”, LEA,pp.183-200 .
- T. M. Duffy, J. Lowyck, D. H. Jonassen ( 1991 ) : “ Designing Environment for Constructive Learning ”, NATO ASI Senes Vol.F105, Springer-Verlag.
- H. G. Widdowson ( 1977 ) : “ Teaching Language as Communication ”, Oxford University Press.
- Susumu Kuno ( 1989 ) : “Danwa - no - Bunpou( Grammar of Discours )”, Daisyukan-Syoten.
- Nobutaka Kato, Yi Liu, Tomonori Manome, Hisayuki Kanda, Makoto Itami, Kohji Itoh ( 1997 ) : “ Use of Situation-Functional Indices for Diagnosis and Dialogue Database Retrieval in a Learning Environment for Japanese as Second Language ”, Proceedings of AIED '97, pp.247-254.