

SGS: A SYSTEM FOR MECHANICAL GENERATION OF JAPANESE SENTENCES

Taisuke Sato

Electrotechnical Laboratory
Ibaraki, Japan

SGS is a compact sentence generation system. Inputs are the frames and specifications of a sentence. Programs attached to context free rules carry out the generation task. Output is a surface sentence with an associated derivation tree.

Introduction

A sentence generation process can be considered to be a process starting from non-linear meaning structures and ending in a linear structure, i.e. a sentence. Because meaning structures reflect the speaker's intension and a speaker can easily produce a sentence realizing his intension, one tends to think of sentence generation as an easy task.

Famous AI systems including SHRDLU, often adopt a fill-in-the-blank method to generate answering sentences. Efforts are concentrated on other tasks, such as sentence understanding, planning, deduction, and so on.

Although study of sentence generation has not been receiving much attention, it is valuable for the following reasons:

- 1) to develop a tool which enables a user to understand what has been understood by an intelligent system.
- 2) to build a machine translation system.
- 3) to develop a theory of knowledge representation. If some formalism of knowledge representation is to be valid, it must be readable. In other words, it must be easily transformed into sentences. And this readability is checked by means of sentence generation.
- 4) to verify correctness of the various linguistic theories from a computational linguistic point of view.

SGS is an experimental sentence generation system, the inputs of which are frames representing some meaning. It generates a Japanese sentence with the help of a user-supplied dictionary and grammar. The generation process of the SGS is top-down with backtracking. The result is a surface sentence with its derivation tree. This system does not generate sentences at random but carefully generate one sentence obeying the user's control information which is given in advance of the generation process.

In computational study of sentence generation, building a system is one facet of the study. The other facet is the extraction of

suitable laws for a computer from linguistic phenomena. Therefore, this paper first describes the overall organization of SGS, secondly explains the linguistic structure of Japanese with which SGS tries to deal, and lastly gives examples of sentence generation.

System Organization

SGS is written in ETL-LISP and consists of about 1000 line source statements. To actually produce a sentence, it needs three kinds of input, a dictionary, and a grammar. Accounts are given in order.

Inputs

A sentence generation corresponds to the speaker's speech process. Accordingly, if a sentence of good quality is needed, many factors of a speaker should be incorporated. However we restrict ourselves to treating only the syntactic and semantic factors. The pragmatic factors remained as future problems.

Conceptually, factors considered here are separated into two categories. One is the factors governing the intra-sentential phenomena, which determines the cognitive meaning of a sentence, and is stated in terms of phrase structures, transformation, various features and the like. Inputs belonging to this category are the frames representing cognitive meaning of the sentence to be generated, and the syntactic category (e.g. SS ---Simple Sentence) of the sentence. In this paper, frames in examples are supplied by the Japanese language understanding system EXPLUS.¹⁴

The other category is factors governing inter-sentential phenomena related to "topic and comments", such as the distinction between "wa and ga" e.t.c.. These factors reflect a speaker's views. They can be treated by specifying the arrangement of noun phrases or the surface subject of the sentence. For example, if one wants to put emphasis on a certain noun whose deep case is THEME, specifying (S-SUBJ = THEME) may compel the system to derive a passive sentence whose surface subject is the specified noun. Therefore, such specifications work as conditions on the sentence or control information for the generative process.

In summary, frames, a syntactic category, and conditions on a sentence reflecting a speaker's views comprise the inputs of SGS. Given these inputs, SGS tries to generate a sentence of the specified syntactic category from the frames by considering the given

conditions.

Figure 1 is an example of an input frame. This frame represents the fact that HANAKO BUYS A BOOK. The REL-TM slot designates relative-time relation to other facts. The SF slot designates semantic features of the predicate KAW-D (BUY). CACT(causal actant), THEME are the deep cases of KAW-U.

```
((IDENT = (P000006 PROPOSITION))
 (LINK
  (THEME (N000007 HON))
  (CACT (N000008 HANAKO))
  (REL-TM (P000005 PROPOSITION)))
 (SELF = (a ITU-DOKO with REL-TM) (a MODALITY))
 (REL-TM = (SORE-WA (P000005 PROPOSITION) MAE))
 (SF = %VASPD %UT2)
 (PREDICATE = KAW-U VERB)
 (CACT = (N000008 HANAKO))
 (THEME = (N000007 HON)))
```

Figure 1. Example of Frame.

Grammar

Grammar in SGS refers to the collection of context free rules augmented by LISP programs. The role of the grammar is to systematically convert input frames into small trees, then combine and transform them while making sure of the grammatical correctness of the generated trees.

It is not necessary for tree structures to accompany sentence generation (McDonald's system doesn't use tree structures), but sentence generation via tree structures has many advantages. First of all, linguistic knowledge based on transformational theory can be easily implemented in a computer. Linguistic concepts such as subject, object, scope of quantifier, deletion, raising, e.t.c., are all related to tree structure. Also, organizing the system as a tree manipulation system is a good way to keep its clarity and is helpful in debugging the grammar. Suggestive information to improve grammar could be obtained by tracing intermediate trees. Moreover, context free rules to construct a derivation tree assures, to some extent, the grammatical correctness of the generated sentence. The form of a syntactic rule is:

```
(<category> <descendants> <P1> <P2>)
```

A rule has four fields. <category> and <descendants> form a context free rule: <category> ⇒ <descendants>.

<P1> is a LISP program. It is applied to the frames which should be realized as a sentence of the <category>. It divides the frames into subframes corresponding to each <descendants> considering the attached conditions.

<P2> is also a LISP program. It is invoked after the completion of <descendants> subtrees. Its role is to look at the subtrees and make sure of their grammatical correctness. Transformation is added to the subtrees as necessary. Finally, <P2> returns a partial derivation tree whose top node is <category>. The rule invocation mechanism is explained later.

Dictionary

A lexical item in dictionary describes the knowledge of each word. As for the predicate, a name, a surface expression, semantic features, deep cases and their semantic features are included in its description. Similar items are included in the noun's frame. The form of an item is:

```
(<name> <category> <P1> <unit>)
```

<name> and <category> are keys for searching the dictionary. In the case of HON (a book), the <name> is HON, the <category> is noun.

<P1> is a LISP program to check conditions for lexical insertion.

<unit> is a frame depicting linguistic knowledge of a word. World knowledge can also be stored in <unit>.

The description of a lexical item is at a concrete level. Neither lexical decomposition nor word description by primitives is adopted. Although, with respect to verbs, Japanese has a rather systemic way of deriving new words from a basic word (for example, from TOB-U (to fly), TOB-ASU (to make something fly) or TOB-ERU (can fly) are derived.), studies in relations among the lexical items seems not to be advanced enough for use in a computer at present.

Generation mechanism

There are many methods to generate sentences. The fill-in-the-blank method is easiest. McDonald's system^{8,9} derives a sentence directly from source data. BABEL^{4,5} derives a sentence indirectly using discrimination nets and a syntax net.

As stated previously, SGS generate a sentence via tree structures. Initially, SGS receives an ordered triple from a user. Its form is:

```
<category, input-frames, conditions on the sentence>
```

The system regards the ordered triple as a goal. It says "from the input-frames, generate a sentence of the category that satisfies the conditions". After pushing this triple to the bottom of the stack, the system starts the generation process described below.

step 1: lexical insertion

Look at the top of the stack. Let this triple be category A, frame Fr-A, condition Cond-A. Collect lexical items from the dictionary that match Fr-A and satisfy Cond-A. If no item is found, Go to step 2. Else, choose one of the items and return it. Because back-track may occur in later process, preserve the unchosen items. Remove the top element from the stack. Go to step 1.

step 2: subgoal expansion downward

If subtrees under category A are completed, go to step 3. Else collect rules of the form A descendents P1 P2 from the grammar. Select one of them. Suppose the selected one is <A (B C) P1 P2>. Execute program P1 to create the subgoals. P1 tries to divide Fr-A into Fr-B and Fr-C. P1 also converts Cond-A to Cond-B and Cond-C respectively. If this division is successful, push the resulting subgoals <B Fr-B Cond-B> and <C Fr-C Cond-C> onto the stack. Go to step 1. If division is unsuccessful, try another rule. If all the tried rules fail, start back-tracking.

step 3: tree building upward

This step treats the case where subtrees under category A are completed. Execute program P2 in the rule <A descendents P1 P2> which was used to divide Fr-A at step 2. P2 tries to confirm the grammatical correctness of the completed subtrees using interpretation of them. If one of them is found to be ungrammatical, start back-tracking. Else transform them as necessary and provide data for later interpretation of the completed tree. Combine the category A and subtrees to complete the partial derivation tree corresponding to the goal <A, Fr-A, Cond-A> on the top of the stack. Remove this triple from the stack. If the stack is empty, collect the terminals of the tree in left-to-right order, give morphological inflection to the sequence of terminals and print them. Otherwise, go to step 1.

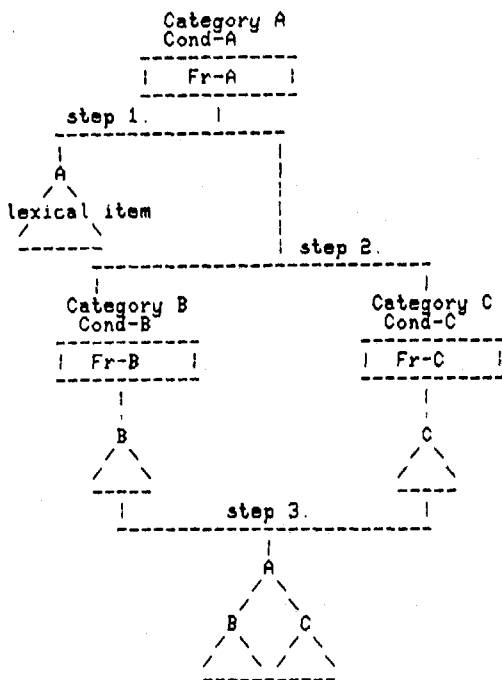


Figure 2 Generation Process

Simplified Syntax of Japanese

This section gives a brief account of the simplified Japanese which SGS tries to deal with.

A Japanese simple sentence consists of three parts, as is shown below. It is important to notice that these parts assume different functionalities.

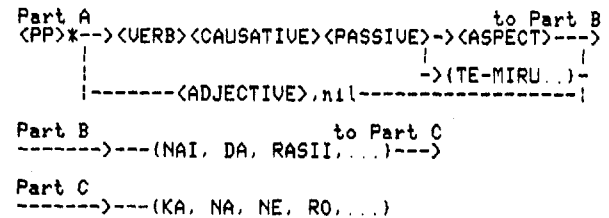


Figure 3. Simplified Syntax of Japanese

The first part, A, expresses epistemic meaning of the sentence. It begins with several propositional phrases (typically two or three) in unspecified order. A prepositional phrase is derived by the rule <PP => <NP><P>, where <NP> is a noun phrase, and <P> is a post positional particle. Particles belonging to <P> are GA, NO, NI, WO, DE, etc.. They work as surface case markers.

After a sequence of <PP>s, there comes a verb, an adjective, or nothing. A verb can be followed by SERU·SASERU (causative particles) or RERU RARERU (particles of passive or spontaneity etc.). These particles are connected to a verb so tightly that they work as a single word. Words for <ASPECT>, TE-MIRU(to try), TE-AGERU (to indicate a speaker's attitude to the hearer in which a speaker kindly does something for the hearer), TE-KURERU(opposite to TE-AGERU), etc. are the last constituents of part A. These are all verbs.

The second part, B, indicates a speaker's attitude to the proposition expressed by part A. This part contains DA(affirmative), NAI(negative), DAROU(guess), RASII(conjecture), etc.. These are all particles.

Expressions in the last part, C, are meant to cause some effect on the hearer. Among them are KA(interrogative), NA(prohibition), NE(suggestion), RO(imperative), etc..

A predicate(verb, adjective) has a case structure. For example, OK-U (put) has three cases: CACT(causal actant), THEME, and LOCUS. Each case is accompanied by specific particles. CACT and GA, THEME and WO, LOCUS and NI or DE are usually used in pairs. The case system is a basic linguistic structure in itself, but the primary objective of SGS is not the study of case system in Japanese, so SGS utilizes the case system of EXPLUS.

Syntactic rules governing the connection of particles following a predicate are said to be described by a regular grammar.

As for tense representation, TA is used to indicate the past or perfect tense. TA can be inserted between either part A and part B, or

part B and tense systems are discussed in the following sections.

A compound sentence is composed of simple sentences. A relative clause in Japanese is derived by the rule <NP> ⇒ <S><NP>. This rule yields a left branching structure peculiar to Japanese in contrast to English. In this paper an example of a sentence using a relative clause is shown with discussions. However, sentences with coordinate structures are not treated.

From a transformational stand point, embedding structures are important. A causative or passive sentence is typical of embedded structures. A generation example of a causative passive sentence is shown later. But how a passive and causative sentence is derived from the initial structure is not definitely solved.

Aspect^{1,2}

In order to achieve temporal representation, treatment of tense and aspect is inevitable. First we discuss the Japanese aspect system which brings a lot of insights useful to computational linguistics.

The basic role of aspectual representation is the distinction between perfect and non-perfect. It seems to be common to many languages. However, actual languages provide mechanisms for aspectual representation developed beyond this distinction.

In Japanese, many types of aspects are realized by using aspectual particles following a verb. TE-IRU and TE-SIMAU are most typical. For instance, YON-DE-IRU (YON is a contracted form of the verb YOM-U (to read)) means the repeat of reading or the experience of reading. YON-DE-SIMAT-TE-IRU means being in the state after the achievement of reading. Several primitive aspects are shown in fig. 4.

aspect-	stative --simple, resultative, progressive ..nil, TE-IRU. inchoative ..SI-KAKERU, SI-HAZIMERU, etc.. completive ..TE-SIMAU, SI-OUARU, etc.. others ..TE-ARU, TE-YUKU, TE-KURU, etc..
predicate	+stative +durative ..ADJECTIVE, DA, etc. -durative ..NIRU(resemble),etc. -stative cross classification by +durative,+resultative,etc.

Figure 4. Classification of Aspects and Predicates

There are stative, inchoative, completive, and other aspects. The stative aspect is subclassified into three subclasses. TE-IRU performs an important role in establishing these subclasses.

Verbs in Japanese are classified according to the aspectual meaning of the combination of the verb and aspectual particles. As a result, aspect features are assigned to a predicate and an aspectual particle. For example, [-durative, +resultative] is assigned to OK-U. [+stative, +durative] is assigned to an adjective or a

cupulative expression DA, and so on. With regard to particles, [+stative], [+completive] are assigned to TE-IRU and TE-SIMAU respectively.

Once aspect features are assigned to the predicate and the particles, an interpretation of the aspect of a composite predicate is mechanically deduced by looking only at the aspect features of each constituent. The aspect of YON-DE-IRU, for example, is obtained in such a way that the aspect features of YOM-U (read) and TE-IRU are examined first. YOM-U has [-stative, +durative, +resultative] and TE-IRU has [+stative]. Then the features are synthesized in obedience to 'synthesizing rules of aspects'. In this case the result is [+durative, +resultative, +stative]. It allows two interpretations, which is compatible with the aspectual ambiguity of YON-DE-IRU. One interpretation, based on the combination [+durative, +stative], is the progressive interpretation---being in the state of reading. The other interpretation, based on [+resultative, +stative], is the experiencing interpretation--- being in the state of after reading. These aspectual ambiguities are resolved by context or adverbials. Similarly, the aspect of YON-DE-SIMAT-TE-IRU is obtained in the same way.

It is easy to see the advantage of 'aspect description by aspect features'. It enables us to treat the (Japanese) aspect mechanically in both directions -- sentence understanding and sentence generation. However, though a great deal of progress has been made in the study of Japanese aspects, we have not yet devised a satisfactory system for aspect description by aspect features.

Tense^{1,2}

It is well known that TA stands for not only past tense but also the speaker's confirmation, recollection, or immediate requirement. Consequently, we can not simply say that TA indicates past tense. Instead there are a number of evidences suggesting that TA indicates the perfect as well. As will be explained in the following, treating TA as a perfect-indicator leads to a succinct description of tense interpretation in Japanese. This fact itself, in the author's opinion, is the strongest evidence for TA as a perfect-indicator. [+perfect], therefore, is assigned to TA. It is also assigned to a predicate accompanying TA. If a predicate does not accompany TA, [-perfect] is assigned. Some definitions are needed before stating tense interpretation in Japanese.

Definition: speech time is the time when a speaker speaks, and event time is the time occupied by the events(facts) referred to by a sentence or a clause.

With this definition, the principle of tense interpretation in Japanese is stated as follows.

A sentence of a clause containing a predicate of +perfect(-perfect) refers to the events or facts previous(not previous) to the standard time.

The standard time of a simple sentence or a main clause is the speech time. The standard time of a subordinate clause is the event time referred to by the main clause.

In short, TA asserts something has occurred previously. Detailed tense interpretation using the aspect feature 'stative' is summarized in figure Fig. 5 which is hereafter called 'the principle'.

aspect of predicate	+perfect	interpretation at the standard time(*present)
+stative	-perfect	present state
	+perfect	past state
-stative	-perfect	present or future action truth, habit
	+perfect	past action, event habit

Figure 5. Principle of Tense Interpretation

The principle is applicable to any simple sentence and the majority of complex sentences. However some complex sentence has exceptional tense interpretation. Consider the next sentence in which the conjunctive TOKI is used.

KAKI-WO TABE-TA TOKI KANE-GA NAT-TA.
(a persimmon)(ate) (a bell)(rang)
When I ate a persimmon, a bell rang.

According to the principle, TA of TABE-TA assures that eating-a-persimmon precedes bell-ringing. But, unfortunately, such is not the case. The fact implied by the sentence is the simultaneity of eating-a-persimmon and bell-ringing.

Such an exception may be ascribed to the peculiarity of the conjunctive TOKI. Since TOKI is also a noun and means time. TOKI used as a conjunctive is apt to connote 'at the time when'. Exceptions to tense interpretation seem to depend on the conjunctive in the case of an adverbial clause, or the head noun in the case of a relative clause. Therefore case studies of tense interpretation are needed.

Tense interpretation of the sentence type S1-- conj --S2 concerning Japanese tense conjunctives TOKI(when), MAE(before), ATO(after) is summarized in Fig. 6.

S1 is a subordinate clause. S2 is a main clause. The aspect feature +-stative is a feature belonging to the predicate of S2. 'applicable' means that the principle is applicable. 'simultaneous' means that the tense interpretation is exceptional and the simultaneity of the events referred to by S1 and S2.

In the case of relative clauses, a tense interpretation table like the above can be similarly constructed, but the situation is worse in the case of adverbial clauses. There

conj.	S1	S2	interpretation	
TOKI	-stative	-perfect	+perfect	simultaneous or applicable
		+perfect	-perfect	
	+stative	-perfect	+perfect	simultaneous
		+perfect	-perfect	
MAE	-stative	-perfect	+perfect	applicable
			-perfect	
ATO	-stative	+perfect	+perfect	applicable
			-perfect	

Figure 6. Conjunctives and Tense Interpretation

also exist complex sentences requiring tense interpretation opposite to the principle.

TAROU-GA TABE-TA KEIKI-WA HANAKO-GA TUKUT-TA.
(TAROU--name)(ate)(a cake)(HANAKO--name)(made)
The cake that TAROU ate was made by HANAKO.

The main clauses is HANAKO-GA TUKUT-TA (HANAKO made a cake). The relative clause is TAROU-GA TABE-TA(TAROU ate the cake). Both clauses include TA, so the prediction by the principle is that the event TAROU-GA TABE-TA precedes the event HANAKO-GA TUKUT-TA, which is exactly opposite to usual tense interpretation.

Because of these difficulties, SGS did not go far with respect to aspect and tense interpretation. Obviously further investigation from a linguistic point of view is needed for mechanical aspect-tense interpretation.

Generation Examples

Relative Clause

There are two types of relative clause. One is the TAROU-GA TABE-TA KEIKI(the cake which TAROU ate) type. The other is the TAROU-GA KEIKI-WO TABE-TA ZIZITU(the fact that TAROU ate a cake) type.

The example shown is the former type. Meaning structures consist of two propositional frames P000001 and P000002. Note that they have a common filler (N000002.ITA).

Initially specified are the top category SS, the arrangement of propositional phrases--first THEME then LOCUS, and the surface subject --THEME. These inputs are goals saying "generate a sentence from the frames shown in figure 7. As to the sentence, its category must be SS(simple sentence), its surface subject must be THEME--ITA(a board), and THEME must be to the left of LOCUS".

On receiving these inputs, the system starts rule invocations. The invoked rule

selects a frame suitable for a main clause. Priority of the selection is given to the frame which includes a REL-TM(relative time) slot filled with "HATUWA"(speech time). In this example, P000001 is selected. It states that TAROU-GA ITA-WO TATEKAKE-RU(TAROU leaned a board somewhere). P000001 being selected, the system continues invoking rules in order to translate P000001 into a main clause.

During the course of rule invocations, the generation process reaches the stage where the THEME slot is treated. Because the THEME slot and its filler--(N000002.ITA), are always supposed to correspond to a noun phrase, rules of the form <NP>=>... are invoked one by one.

As (N000002.ITA) is shared with another frame, P000002, which states that HANAKO-GA ITA-WO OI-TA(HANAKO put a board), <NP>=><SS><NP>, a rule for a relative clause, eventually is invoked. It produces a relative clause--HANAKO-GA OI-TA ITA(a board which HANAKO put

somewhere). It first builds a tree for the sentence HANAKO-GA ITA-WO OI-TA from P000002 and completes the relative clause by moving the position of ITA to the end of the sentence. Generally speaking, complex noun phrase restrictions should be considered, but they do not work here. After the completion of the relative clause concerning (N000002.ITA) with a corresponding derivation tree, SGS tries to complete the main clause, but, since the rule invoked for the main clause allows only CACT--TAROU as a surface subject, it can not satisfy one of the initial goals (S-SUBJ = THEME). So backtrack occurs.

Finally, the alternative rule <SK>=><SK><RARE1> is invoked. It generates a passive sentence whose subject is THEME--ITA, and the rest of the specifications are also satisfied. '-*-' in the derivation tree indicates a non-existent filler of the obligatory case in the given frame.

A passive sentence treated by SGS is 'a pure passive sentence' which does have a counter part in English. There is also another type called 'an adversitive passive sentence'. This type is too subtle to treat mechanically. Therefore we consider only pure passive sentence and the rules for them.

Causative Passive Sentence

Japanese causative sentences, which are identified by the occurrence of VERB + SERU. SASERU, often admit two types of interpretation. Consider the next sentence.

```

@MTM==
  ((IDENT = (P000002 PROPOSITION))
   (LINK
    (THEME (N000002 . ITA))
    (CACT (N000003 HANAKO))
    (REL-PTM (P000001 PROPOSITION)))
   (SELF = (a ITU-DOKO with REL-PTM) (a MODALITY))
   (REL-PTM = (KANRYOU (SORE-WA (P000001 PROPOSITION) TOKI)))
   (SF = %VASPC %UT1)
   (PREDICATE = OK-U VERB)
   (CACT = (N000003 HANAKO))
   (THEME = (N000002 ITA)))
  ((IDENT = (P000001 PROPOSITION))
   (LINK = (THEME (N000002 . ITA)) (CACT (N000001 . TAROU)))
   (SELF = (a ITU-DOKO with REL-TM) (a MODALITY))
   (REL-TM = (KANRYOU (SORE-WA "HATUWA" TOKI)))
   (SF = %VASPC %UT1)
   (PREDICATE = TATEKAKE-RU VERB)
   (CACT = (N000001 TAROU))
   (THEME = (N000002 . ITA))))
@STM==
  ((IDENT = (N000003 HANAKO)) (SELF = (a HITO)) (SF = %ANIMAL))
  ((IDENT = (N000002 ITA))
   (SELF = (a SYAHEIBUTU) (a HEIMEN))
   (SF = %ARTOBJ))
  ((IDENT = (N000001 TAROU)) (SELF = (a HITO)) (SF = %ANIMAL))

```

```

-- Initial-CATEG --
$ SS
-- Initial-COND --
$ (S-SUBJ = THEME)(SPAN-SEQ = THEME LOCUS)()

```

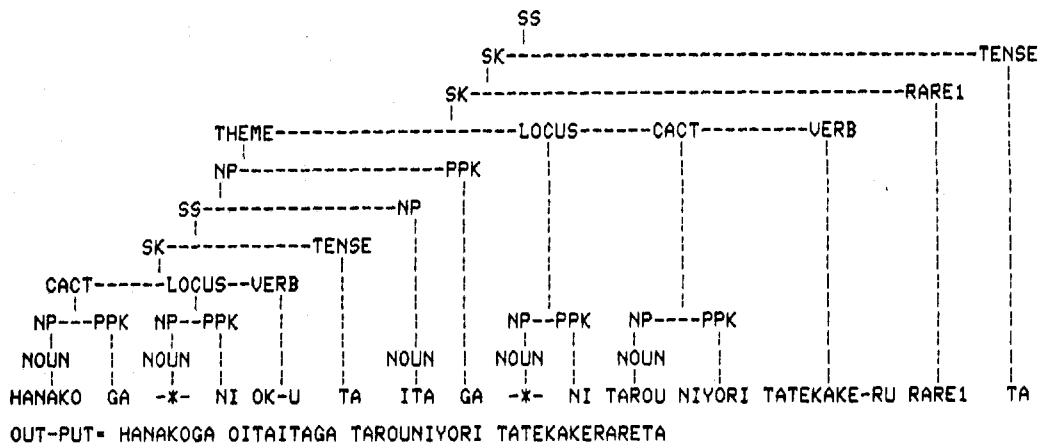


Figure 7. Sentence with a Relative Clause.

```

@MTM==
((IDENT = (P000068 . PROPOSITION))
 (LINK = (THEME (N000140 . HASIGO)) (CACT (N000141 . HANAKO)))
 (SELF = (a ITU-DOKO) (a MODALITY with TENSE))
 (TENSE = KANRYOU)
 (SF = %VASPC %UT1)
 (PREDICATE = TATEKAKE-RU VERB)
 (CACT = (N000141 . HANAKO))
 (THEME = (N000140 . HASIGO)))
((IDENT = (P000067 . PROPOSITION))
 (LINK = (THEME (P000068 . PROPOSITION)) (REL-TM (P000068 . PROPOSITION)))
 (SELF = (a ITU-DOKO with REL-TM) (a MODALITY))
 (REL-TM = (SORE-WA (P000068 . PROPOSITION) MAE))
 (TENSE = (KANRYOU (SORE-WA "HATUWA" TOKI)))
 (SF = %VASPC %UT2)
 (PREDICATE = SASE-RU VERB)
 (THEME = (P000068 . PROPOSITION)))
@STM==
((IDENT = (N000141 . HANAKO)) (SELF = (a HITO)) (SF = %ANIMAL))
((IDENT = (N000140 . HASIGO)) (SELF = (a BUTTAI)) (SF = %ARTOBJ))

```

```

-- Initial-CATEG --
$ SS
-- Initial-COND --
$ (S-SUBJ = THEME)()

```

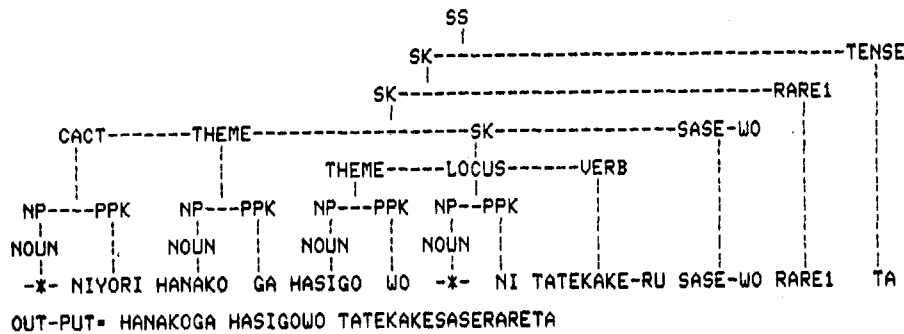


Figure 8. Causative Passive Sentence.

TAROU-GA HANAKO-WO YUKA-SERU.
(to go)

One interpretation is that TAROU forces HANAKO to go. The other is that TAROU permits HANAKO to go. Ambiguities can be resolved by adverbials or context. These ambiguities bring difficulties to the treatment of causative sentences, but, for simplicity, SGS deals with only the former type.

The example above is a causative-passive sentence. User's specifications are of the category SS and (S-SUBJ = THEME). The initial meaning structures consist of two propositional frames. The generation process begins by choosing a HATUWA frame to serve as an origin of time relations in the given frames. The chosen frame, P000067, includes a predicate slot containing SASERU. It will produce a causative sentence.

While SASERU is a causative particle, it behaves as a verb in the deep level. It is a verb which takes a sentential object whose case is THEME. Therefore the invoked rule responsible for completing a causative sentence searches for a sentential object. P000068 is the frame for a sentential object. It states:

HANAKO-GA HASIGO-WO TATEKAKE-TA.
(a ladder) (leaned)
HANAKO leaned a ladder.

After the completion of the derivation tree and the (embedded) sentence corresponding to P000068, the rule mentioned above notices the tense of P000068 as being +perfect. This feature would entail an occurrence of TA in front of SASERU on the surface level. But word order such as ...TA SASERU... is ungrammatical so TA is suppressed.

A causative sentence corresponding to P000067 is built by raising CACT of the embedded sentence. The raised CACT--HANAKO changes to THEME. The resulting tree structure is, roughly speaking, [-*- HANAKO [HASIGO TATEKAKE-RU] TA]. The symbol -*- means non-existent filler.

Owing to this structure, a passive sentence whose subject is THEME--HANAKO can be derived and (S-SUBJ = THEME)--initial goal-- is satisfied.

Conclusion

Sentence generation is a basic task for an intelligent system, such as a consultant system or a Q.A. system, e.t.c.. SGS, though it is far from being satisfactory, is one step closer to an intelligent sentence generation system. The next step should be manifold. SGS admits various improvements.

During the generation process, diverse messages are exchanged between invoked rules so that messages tend to get out of control. Greater regulation is needed.

As for the dictionary, it would be interesting to incorporate 'lexical decomposition'. Introducing 'lexical decomposition' can be helpful in organizing lexical items in a dictionary. However it requires a more refined method of lexical insertion.

Linguistic knowledge should be thoroughly investigated and digested. Though the aspect-tense system in Japanese has been investigated to some extent, it is not obvious whether the description of aspect-tense system by features is sufficient to represent temporal knowledge.

Presently, SGS lacks the ability to continuously produce sentences. In order to form a paragraph the problem of coreference mechanism

must be solved. Japanese is so rich in ellipsis it is necessary to reveal and implement the ellipsis system.

ACKNOWLEDGEMENT: The author is grateful to Mr. Tanaka, Chief of Machine Inference Section of Electrotechnical Laboratory and, other members of the section, for helpful discussions.

REFERENCES:

- [1] Ota Akira: "Comparison of English and Japanese, with special Reference to Tense and Aspect", Studies in English Linguistics, Asahi Press, 1972.
- [2] Ota Akira: "Tense Correlations in English and Japanese", Studies in English Linguistics, Asahi Press, 1973.
- [3] Huber, F.: "On the Generation of English Sentence", IEEE Trans. of Computers, 25:90-91, 1976.
- [4] Goldman, N.M.: "Computer Generation of Natural Language From a Deep Conceptual Base", Stanford AIM-247, Jan. 1974.
- [5] Goldman, N.M.: "Sentence Paraphrasing from a Conceptual Base", Comm. Assoc. for Computer Machinery, 2, 18, 1975, 96-106. Academic Press, 1975, 41-58.
- [6] Hutchins, W.J.: "The generation of syntactic structures from a semantic base", North-Holland, 1971.
- [7] Kuno Susumu: The Structure of Japanese Language, MIT press, 1973.
- [8] McDonald, D.: "Preliminary Report on a Program for Generating Natural Language", IJCAI4, 1975, 401-405.
- [9] McDonald, D.: "A Framework for Generation Grammars for Interactive Computer Programs", AJCL, Microfiche 33:4, 1975.
- [10] Schank, R.C.: "Conceptual Information Processing", North-Holland, 1975.
- [11] Self, J.: "Computer generation of Sentences by Systemic Grammar", AJCL, Vol.12-5, Microfiche 29, 1975.
- [12] Shapiro, S.C.: "Generation as Parsing from a Network into a Linear String", AJCL, Microfiche 33 : 45, 1976.
- [13] Simmons, R. and Slocum, J.: "Generating English Discourse from Semantic Networks", CACM, Vol.15, No.10, 1972.
- [14] Tanaka et al.: "EXPLUS-A Semantic Parsing System for Japanese Sentences", Third USA-JAPAN Computer conference, 1978.