# A MACHINE TRANSLATION SYSTEM FROM JAPANESE INTO ENGLISH
## — ANOTHER PERSPECTIVE OF MT SYSTEMS —

M. Nagao, J. Tsujii, K. Mitamura, H. Hirakawa, M. Kume

Department of Electrical Engineering
Kyoto University
Sakyo, Kyoto, 606, JAPAN

Summary

A machine translation system from Japanese into English is described. The system aims at translation of computer manuals, and basically follows to the transfer approach. The design principles of the system are discussed in detail, together with the overall constructions of the system. Especially, the effectiveness of lexicon-based procedures, i.e. lexicon-based analysis, transfer, and synthesis, is emphasized. Most of the linguistic phenomena are treated by using lexical descriptions and lexical rules, instead of by general syntactic rules. Because Japanese and English belong to quite different language families, much more structural transfers are necessary than in other MT systems among European languages. Special cares have been paid for designing the transfer component. Some translation results are also given to illustrate the current abilities of the system.

## 1. Introduction

This paper is the first progress report of a machine translation system from Japanese into English being developed at Kyoto University. The project currently aims at the translation of computer manuals, in which vocabulary is rather limited and less ambiguous than in other subject fields. However, this system is a good example of MT systems whose SL's and TL's belong to quite different language families, in which a lot of interesting problems have arisen that have been concealed in the systems whose language pairs are rather close in language families. We will discuss in the paper some of the design principles without referring to the detailed linguistic phenomena.

The system has been implemented on FACOM M-200 (Kyoto University Computing Center) mostly by LISP. Only exception is the morphological analysis of Japanese, which is done by PL/1 program.

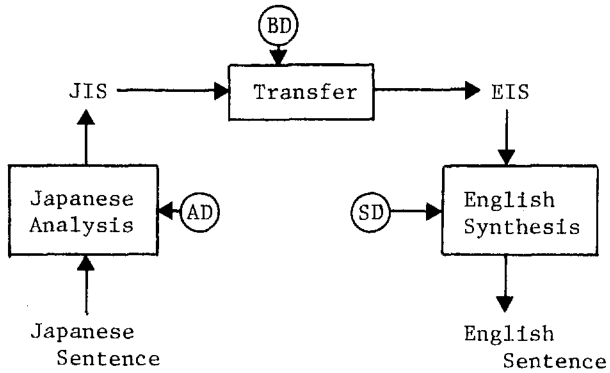
The system basically follows to 'transfer approach' advocated by several other groups such as TAUM, GETA etc.(1) The overall system consists of the three major components; Japanese analysis, transfer, and English synthesis components as shown in Fig. 1. The system is based on several guiding principles. Among these, the followings would distinguish our system from the other MT systems.

1. It is highly lexicon-driven. Every component including analysis, transfer and synthesis components is highly dependent on lixical descriptions of individual words. In other words, most of the linguistic phenomena are treated by lexical descriptions and lexical rules, instead of general syntactic rules such as 'structure dependent rules' in Chomskian grammar. We completely agree with J. Bresnan, an MIT linguist, when she claimed as follows: (2)

> 'Finally, I assume that it is easier for us to look something up than it is to compute it. It does in fact appear that our lexical capacity — the long-term capability to remember lexical information — is very large.'

2. The approach becomes closer to the interlingual approach. Because Japanese structures can be adequately captured by dependency structures based on case notions, we adopted this structure as the intermediate representation for Japanese. On the other hand, the structures from which synthesis of English will start are ordinary phrase structures. It is well known that dependency structures require semantically deeper analyses than usual phrase structures. Therefore, our approach becomes closer to the interlingual approach, and even undistinguishable with it in some cases. Especially, because the two languages have quite different systems for expressing tenses, modals, aspects etc., these expressions are analyzed into much deeper levels, that is, almost the interlingual level. Considering the fact that the two languages belong to quite different language families, our approach seems to be inevitable.

3. Stereotyped or semi-stereotyped expressions found in computer manuals are effectively utilized. Stereotyped expressions here mean not only idioms in a usual sense, but also certain stylistic prototypes which can often be found in manuals. Special cares have been taken to utilize them effectively in our system.




JIS : Japanese Intermediate Structure (Dependency Structure Based on Cases)
EIS : English Intermediate Structure (Phrase Structure Tree)
AD : Analysis Dictionary for Japanese
BD : Bilingual Dictionary
SD : Synthesis Dictionary for English

Fig. 1 Overall Construction of the MT System

## 2. Japanese Sentence Analysis

The analysis proceeds as follows:

1. morphological analysis
2. segmentation of an input sentence into a set of simple sentence fragments (each fragment contains only one predicative term such as verb, predicative adjective, copula, etc.)
3. recognition of relationships among sentence fragments
4. noun phrase analysis } performed
5. simple sentence analysis } intermixedly

Because Japanese is a typical agglutinative language, many useful sorts of information can be obtained by morphological analysis. It is undoubtedly true in both cases, Japanese analysis and other European language analysis, typically in English analysis, that morphological and syntactic analyses should work co-operatively. However, the co-operation should be done in different ways. Generally speaking, English morphological analysis needs much help from its syntactic analysis. English homograms can rarely be resolved by intra-word processings. Therefore, morphological analysis alone will produce highly ambiguous results in English. Syntactic and even semantic information is required to resolve them. On the contrary, Japanese morphological analysis offers much help to its syntactic analysis. This implies that Japanese morphological analysis can be done in a separate phase with syntactic and other succeeding processings.

Because Japanese morphological analysis is closely related to both the writing system and detailed word inflection rules of Japanese, we shall omit the discussion of this phase, only noting that certain composite expressions are treated in our system as single morphemes. Some examples are shown in Fig. 2. A detailed discussion about this phase can be found in [5].

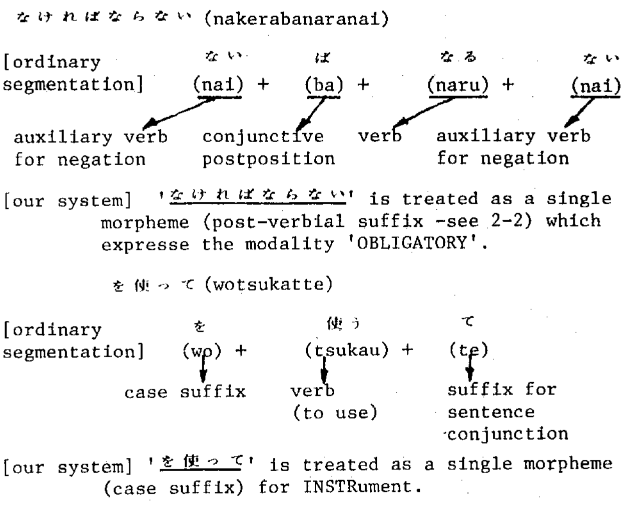
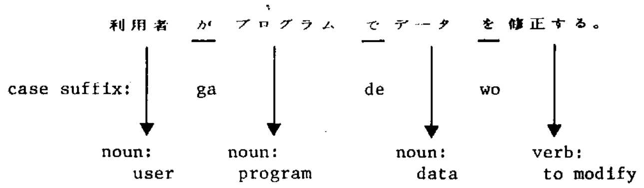



Fig. 2 Examples of Composite Morphemes

### 2-1. Lexicon Based Analysis Procedure for Japanese

In order to discuss the other analysis steps, we have to mention certain syntactic aspects of Japanese. Among those, it should be noted that case relationships between noun phrases and verbs are usually marked by case suffixes attached to noun phrases. An example is shown in Fig. 3.




meaning : (The) user modifies (the) data by (a ) program.

Note : が(ga), で(de), and を(wo) are the case suffixes. In Japanese, noun phrases which bear some grammatical relationships with a verb always precede the verb in a surface sentence.

Fig. 3 Case Suffixes in Japanese

が(ga) usually marks AGENTIVE, を(wo) OBJECTIVE and で(de) INSTRUMENTAL cases, respectively. However, this direct correspondence between surface case suffixes and deep cases may not be preserved in actual sentences. In other words, case suffixes indicate only surface grammatical relationships between noun phrases and a verb, and these grammatical relationships may not coincide with deep semantic cases. We should distinguish them carefully, as C. Fillmore did in English. He tried to set up general rules to relate deep cases with surface grammatical relationships in English. Unfortunately, his model is based on generating sentences and gives us no clue as to how to parse them. Moreover, we observed that, at least in Japanese, this surface and deep correspondence is more or less specific to individual verbs. The same phenomena have been observed in English by J. Bresnan and other linguists.(2) They have treated these phenomena by setting up 'lexical interpretation rules' which are specific to individual verbs, and which translate the surface grammatical structures into deep semantic ones. From computational view points, this fremework leads us to lexicon-based analysis procedures. Instead of general syntactic rules, we describe specific surface-deep mappings for individual verbs in the analysis dictionary, as shown in Fig. 4. One of the main purposes to establish transformation rules was to relate surface structures with deep ones by the rules. In our framework, most of this task is done by surface-deep mappings described in the dictionary. Therefore, a simple pattern matching is sufficient to analyze sentence fragments, that contain only one verb. However, there still remain certain sets of transformations which seem not to be well captured by the surface-deep mappings of individual verbs. We also treat them as lexical rules. We will discuss this point in the next section.

```
修正する

(to modify)

surface pattern :
   1    が     2     で      3     を
       (ga)        (de)          (wo)

deep structure :
   (* MODIFY  (AGENT (1))(INST (2))(OBJ (3)) )
```

In the actual inplementation, sets of semantic restriction are described here.

Fig. 4 A Surface-Deep Mapping

## 2-2. Transformations as Lexical Rules

Transformations treated by our system can be classified into the following categories (Notice that we use here the term 'transformations' in a broader sense than in traditional TG. And also notice that, though 'scrambling' operations are very conspicuous in Japanese which are applied after transformation cycles in traditional TG's, we do not consider them as transformations here, because they can be embodied in pattern matching operations, i.e., pattern matchings without considering orders of elements).

1. Transformations dependent on a set of specified case elements (Fig. 5, Ex. 1) : These correspond to the Fillmore's examples, 'John broke the window with a hammer,' 'A Hammer broke the window,' 'The window broke'.

2. Transformations caused by adverbial suffixes (Fig. 5, Ex. 2) : As shown in Ex. 2, a case suffix can be replaced by an adverbial suffix. Careful investigation reveals that a certain class of case suffixes can be replaced by an adverbial suffix without any traces (TSP1, TSP2 in Ex. 2) and another class of case suffixes cannot be, but just be follwed by an adverbial suffix (TSP3 in Ex.2). In fact, a relative ordering of case suffixes exists and higher case suffixes in the ordering can easily be replaced with an adverbial suffix without any surface traces. Moreover, this relative ordering of case suffixes depends on individual verbs, depending on how intimate a relationship the concept expressed by each noun phrase bears to the action expressed by the verb. We may be able to capture this intimacy hierarchy by setting up several different levels of connections between noun phrases and verbs, as Chomsky does in his X-theory.(3) However, from computational view points, especially from recognition view points, it is convenient to mark in each surface pattern what ordering exists and which case suffixes can be replaced by which adverbial suffixes.

3. Transformations caused by post-verbial expressions (Fig. 5, Ex. 3) : Post-verbial expressions also cause surface pattern transformations. These expressions specify tenses,

SSP : Standard Surface Pattern
TSP : Transformed Surface Pattern

▷Ex.1 (Specified Case Elements)

SSP : 利用者 が プログラム で データ を 修正する。
user (ga) program (de) data (wo) modify
[(The) user modifies (the) data by (the) program.]
TSP : プログラム が データ を 修正する。
program (ga) data (wo) modify
[(The) program modifies (the) data.]

▷Ex.2 (Adverbial Suffixes)

SSP : Same as Ex.1
TSP1: 利用者 も プログラム で データ を 修正する。
user (mo) program (de) data (wo) modify
[(The) user also modifies (the) data by (the) program.]
TSP2: 利用者 が プログラム で データ も 修正する。
user (ga) program (de) data (mo) modify
[(The) user also modifies (the) data by (the) program.]
TSP3: 利用者 が プログラム でも データ を 修正する。
user (ga) program (de-mo) data (wo) modify
[(The) user also modifies (the) data by (the) program.]

Note : も (mo) is an Adverbial Suffixes.

▷Ex.3 (Post-Verbial Expressions)

SSP : Same as Ex.1
TSP : プログラム で データ が 修正してある。
program (de) data (ga) to modify
[(The) data is modified by (the) program.]
* The post-verbial expression 'てある' changes the aspectual feature of 'modify' from 'ACTION' into 'STATE'.

Note : Though the same case elements appear in this Japanese sentence as in TSP in Ex.1, passive construction should be chosen in this case because English passives also change the aspectual feature of the verb.

SSP : Same as Ex.1
TSP : プログラム で データ が 修正される。
program (de) data (ga) modify
[(The) data is modified by the program.]
* The post-verbial expression 'れる' changes the voice of the sentence from 'ACTIVE' into 'PASSIVE'.

▷Ex.4 (Verbal Complement)

SSP : 彼 が 正しい と 思う。
he (ga) be right (to) believe
sentential complement ― complementizer
[(Someone) believes that he is right.]
TSP : 彼 を 正しい と 思う。
he (wo) be right (to) believe
verbal complement ― complementizer
[(Someone) believes him to be right.]

Note : The case suffix 'が (ga)' in SSP shows that 'he' has the direct grammatical relationship 'SUBJECT' with the complement 'be right'. On the other hand, 'を (wo)' in TSP shows that 'he' has the grammatical relationship with the main verb 'believe', but should be semantically interpreted in relation to the verbal complement 'be right'. This interpretation rule is described in the verb dictionary for '思う -to believe'.

▷Ex.5 (Relative Clause)

SSP : Same as Ex.1
TSP1: 利用者 が 修正する データ
user (ga) modify data
[(The) data which (the) user modifies]
TSP2: データ を 修正する 利用者
data (wo) modify user
[(The) user who modifies (the) data]
?TSP3: 利用者 が データ を 修正する プログラム
user (ga) data (wo) modify program
[(The) program by which (the) user modifies (the) data]
TSP4: 利用者 の 修正する データ
user (no) modify data
[(The) data which (the) user modifies]

Note : TSP4 expresses the same as TSP1. However, the case suffix 'が (ga)' is changed into 'の (no)'. This phenomena is observed only in a relativized construction.

Fig. 5 Transformed Patterns

aspects, modals, and voices of sentences. We now have about 50 such post-verbial expressions. Some of them are shown in Table 1, in which * indicates the expression causes transformations. Notice that, though both the post-verbial expressions 'ことができる' and 'できる' give the modality 'POSSIBLE' to the sentences, only 'できる' changes the surface patterns. Also notice that active-passive transformations in Japanese are included in this category.

| | | |
|---|---|---|
| てある (tearu) | ASPECT : ACTION → STATE-2 | * |
| ている (teiru) | ASPECT : ACTION → STATE, PROGRESSIVE | |
| ことができる (kotogadekiru) | MODAL : POSSIBLE | |
| できる (dekiru) | MODAL : POSSIBLE | * |
| れる (reru) | VOICE : ACTIVE → PASSIVE | * |
| られる (rareru) | VOICE : ACTIVE → PASSIVE | * |
| なければならない (nakerebanaranai) | MODAL : OBLIGATORY | |
| てはいけない (tewaikenai) | MODAL : PROHIBITION | |
| ねばならない (nebanaranai) | MODAL : OBLIGATORY | |
| たい (tai) | PERFOMATIVE : DESIRE (to want) | * |
| 必要がある (hitsuyougaaru) | MODAL : NECESSITY | |
| てよい (teyoi) | MODAL : PERMISSION | |
| ておく (teoku) | ASPECT : ACTION → STATE-2 | |

Table 1 Examples of Post-Verbial Suffixes

4. Transformations caused by verbal complements (Fig. 5, Ex. 4) : A certain class of Japanese verbs require verbal complements, as English verbs 'promiss', 'expect', 'believe', 'want' etc. As shown in Ex. 4, certain noun phrases, which bear grammatical relationships to such verbs, should be semantically interpreted in relation to the verbs in the verbal complements. In the standard theory of TG, these phenomena were also treated by general transformation rules such as raising transformations.

5. Transformations in relative clauses (Fig. 5, Ex.5) : Relativization in English is a typical construction which can be adequately explained by structure dependent transformations such as wh-movement rules. However, a relativized construction in Japanese causes not only noun phrase movement but also the other surface transformations as shown in TSP4 of Ex. 5. Moreover, the noun phrases which can be moved are the phrases that are followed by particular case suffixes in the surface patterns. That is, which noun phrases can be moved is dependent on the case suffixes in the surface patterns, and, therefore, dependent on individual verbs.

6. General Transformations : Clefted constructions, for example, also appear in Japanese.

Because the transformations in the above are more or less dependent on individual verbs which govern the transformed structures, we treat them by lexical rules, i.e., we assume that transformations of surface patterns have been done beforehand, and that the transformed patterns are also stored in the individual verb entries in the analysis dictionary.

In the conventional approaches, there are a set of general transformational rules, which will be inversely applied in turn to input sentences, in order to obtain appropriate 'deep' structures. It has been well known that this inverse application of rules results in combinatorial proliferation of possible structures, partly because such rules are not general rules and only applicable to specific classes of verbs. (consider 'promiss him to go' and 'want him to go' example).

Our approach is to avoid such inverse applications of general rules. We regard most of transformation rules as word specific, and assume that pre-applied, already transformed patterns are stored in the individual verb dictionaries. The schematic view of our analysis procedure is shown in Fig. 6. During the analysis, it only selects appropriate surface patterns (transformed or not) from the dictionary and matches them with the input sentences. You may object to us that such a configuration requires a large memory space for the dictionary. However, it is possible to reduce the dictionary size by using macro expressions, if you can classify verbs and decide which transformations are applicable to which verb classes. These macro expressions will be expanded when the dictionary entries containing the macros are retrieved. When you find a spcific verb behaves quite differently from others, you can specify both its surface patterns and transformed patterns directly in the dictionary without using macros. Our approach is: First, we assume that every verb is specific, and exceptional, i.e., it has its own usages and transformed usages and, if we can find some classes of verbs which behave in the same way, then it is possible to generalize them by using macros.

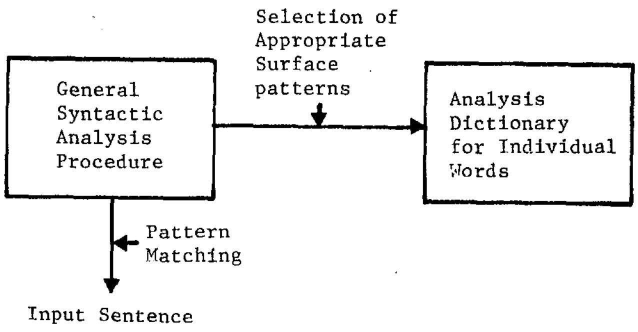



Fig. 6 Schematic View of the Analysis of a Sentence Fragment

In the current version of our system, transformations 1, 2, 3, 4, and 5 can be analyzed. That is, dictionary descriptions for them are prepared (However, because our system is an experimental prototype, the dictionary contains only about 80 verbs).

The information for 1, 2 and 4 is directly coded in the surface patterns. Various transformed patterns for 1 and 4 are stored in the dictionary. As for 2, information as to which one can be replaced by adverbial suffixes are indicated in each surface pattern. As for 3 and 5, each transformed patterns is accompanied with the markers that indicate when the patterns should be used (See 2-3).

## 2-3. Selection of Surface Patterns

As described at the beginning of this chapter, the analysis proceeds in the sequence such as morphological analysis, segmentation of a sentence, recognition of relationships among sentence fragments, and finally, simple sentence and noun phrase analyses. The analysis of simple sentences, the last step, is done by pattern matchings. In this section, we will discuss how to select appropriate (transformed) surface patterns.

At the second step of the analysis, the segmentation step, the input sentence is divided into several sentence fragments so that each of them contains only one predicative term. At the same time, post-verbial suffixes which follow the predicative terms are processed, and the appropriate markers of tenses, aspects, modals, and voices are selected. Moreover, if the suffixes are the ones which cause transformations, the appropriate surface patterns are selected. This selection process is performed in the way similar to Rieger's word exper parser(6) (Fig. 7).

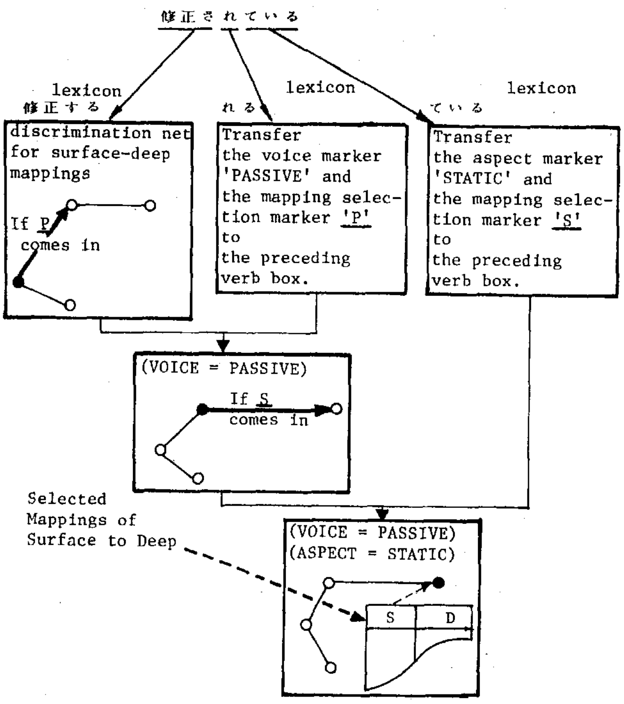
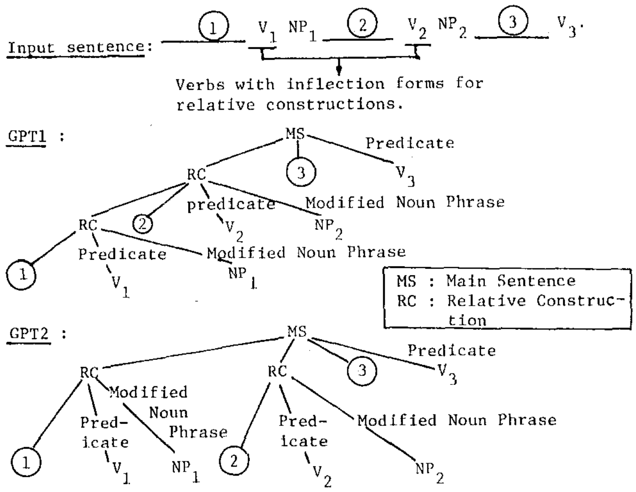



Fig. 7 Selection of Appropriate S-D (Surface-Deep Mappings) Tables for Post-Verbial Suffixes

The third step is to recognize the global structure of the input sentence. The relative clauses, clefted sentences, conjunctions of sentences etc. are recognized at this step, by utilizing the inflection information of each predicative term in the sentence. Generally speaking, several numbers of global structures are produced for an input sentence. Fig. 8




Note : GPT1 and GPT2 represent different global structure for the same input. In GPT1, the first relative clause is embedded in the second. In GPT2, on the other hand, both the two relative clauses are embedded in the main sentence.

Fig. 8 GPT's Which Correpond to the Same Inflection Pattern

shows such an example. The global structure is represented by a tree called GPT (Global Plan Tree), which guides the succeeding analyses. That is, a node of GPT indicates what kind of transformed patterns should be used to analyze the corresponding fragment, and in what oder.

A certain class of transformations can be applied, whenever certain syntactic constructions are found. They do not depend on individual verbs. In relativized constructions, for example, the case suffix 'が' (ga) can be optionally replaced with the other suffix 'の' (no). (Fig. 5, TSP4 in Ex. 5). This rule is not dependent on individual verbs, and moreover, it is not dependent on deep cases. The rule is considered as 'structure dependent'. Because a GPT explicitly indicates by RC nodes where relativized constructions appear, the analysis program transforms the patterns in the dictionary into appropriate forms, when it analyzes fragments governed by a RC node, that is, if a pattern in the dictionary contains the suffix 'が' (ga), the program automatically generates the transformed patterns. Such structure dependent rules are also found in sentence conjunctions, that are similar to the gapping rules in English (sentence conjunctions cannot be analyzed by the current system from the other reasons. We are now designing the procedures for sentence conjunctions).

Because of space considerations we completely omitted the discussions about the noun phrase analysis, the semantic aspects of the processing, the analysis of tenses, modals, aspects and some other troublesome expressions such as adverbial modifiers in Japanese etc. The detailed discussions are found in (5).

## 3. Transfer Step

The transfer is also guided by a lexicon as the analysis procedure is, — in this case, by the bi-lingual dictionary. We will first describe the two structures over which the transfer phase bridges, i.e. intermediate structures for Japanese and English.

### 3-1. Japanese Intermediate Structures — JIS

Japanese intermediate structures produced by the analysis component are basically dependency structures of input sentences, based on case notions. As a usual dependency structure, each node is not labelled by a category symbol like NP, VP, PP etc., but by a word. The word attached to a node is an intermediate word which has a unique entry in the bi-lingual dictionary.
It may happen that a single Japanese surface word corresponds to multiple entries in the bi-lingual dictionary. In these cases, the disambiguation among them is to be done during the analysis phase. However, it may also happen that, during the transfer phase, a single intermediate word should be mapped into several different English words.

Though we claimed that nodes in a JIS was labeled only by an intermediate word that corresponded to a surface Japanese word, there are some exceptions. In order to remedy computational defects of dependency structures, we introduce the other kinds of nodes which do not directly correspond to surface words, but to certain syntactic constructions in Japanese (we call such kinds of nodes 'relation descriptors'). In this sense, our JIS is a mixed form of dependency structures and phrase structures. In principle, our intermediate structures are organized in such a way that a governing node can always determine how to arrange the transferred substructures of its dependents. As will be described in 3-3, a JIS will be evaluated recursively, and the corresponding English intermediate structure will be built up from the bottoms (See Fig. 9).

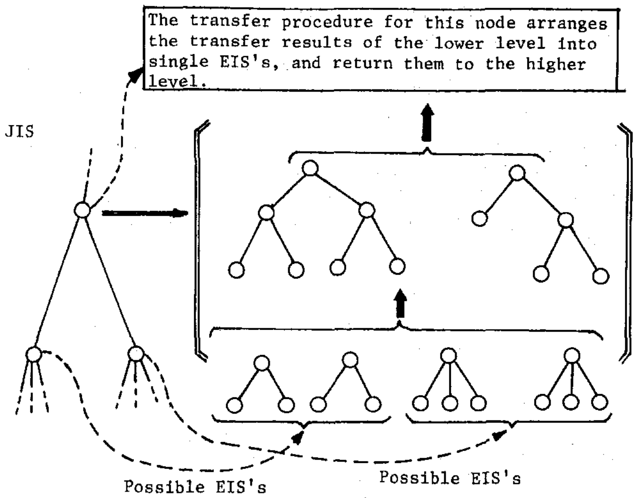
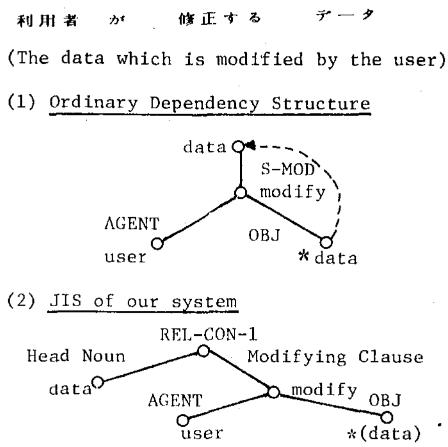



Fig. 9 General View of the JIS-EIS Transfer

In a dependency structure, a noun phrase modified by a relative clause is usually represented by a structure like Fig. 10-(1). However, this structure expresses only implicitly the relationship between the head noun and the modifying clause (* indicates the head noun).




Note : Actually, the node label REL-CON-1 has a unique entry in the Bi-lingual dictionary, which contains the 'transfer procedure' that is responsible for transferring Japanese relative constructions of type 1 into corresponding English ones.

Fig. 10 Comparison of an Ordinary Dependency Structure and JIS

Tree traversing rules would be necessary to recognize that an embedded relative clause exists. Moreover, it is always difficult to determine when to invoke such structure recognition rules, and how to transfer such syntactic structures in the source language into their correspondences in the target. In our JIS, such a syntactic construction is also explicitly marked by 22 node REL-CON-1 in Fig. 10-(2). (Relative clauses in Japanese are subclassified into four different types, according to the relationships between the modified noun and the role which it plays in the modifying clause. Only three of these have direct corresponding relative clause constructions in English). Table 2 shows examples of node labels used in JIS.

| node label | role | node label | role | node label | role |
|---|---|---|---|---|---|
| INST | | REASON | sentence | V-MOD | modifier |
| PURPOSE | | CAUSE | connect- | | for verbs |
| LOC-1 | extrinsic | TIME-SEQ | ives | S-MOD | modifier |
| : | cases | QUALIFY | noun | | for sent. |
| TIME-1 | | QUANTIFY | modifier | CAUSATIVE | |
| : | | REL-CON-1 | relative | surface | |
| CAUSE | | : | const. | words | |

Table 2 Typical Node Labels Used in JIS

Another comment would be necessary on case representation. Many researchers agree that cases are useful in describing linguistic structures, especially semantics of sentences. However, no two agree with each other as to what is the complete set of cases. Our approach is very pragmatic and highly oriented to machine translation. We don't have a 'complete' set of cases in any sense. We always have only a tentative set. If we observe something wrong, we are ready to revise the current set of cases. Moreover, the definition of each case is highly dependent on individual verbs. As discussed in (4 ), we divide the cases into two types (this classification is also dependent on individual verbs). One is the type of cases which are intrinsic to the verb. As to the intrinsic cases, the mappings from Japanese surface to JIS relations are specified in the analysis dictionary, and moreover, the mappings from JIS relations to EIS structures are described in the bi-lingual dictionary (see Fig. 11). To put it in another way, Japanese surface structures that express these cases are mapped into corresponding English structures by the lexical rules in the two dictionaries. There are no general rules which refer to general case notions.

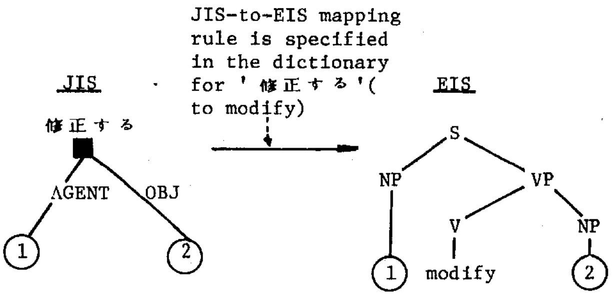



Fig. 11 Structural Transfer for Verbs

The other type of cases, called extrinsic type, is treated differently. For this type of cases, general rules are prepared to transfer them. These rules are independently formulated of individual verbs and show how to express the deep cases in English. Therefore, in contrast to the intrinsic cases, the cases of this type are explicitly expressed by nodes in JIS's (see Fig. 12.) These case labels have their own entries in the bi-lingual dictionary, in which rules for selecting appropriate prepositions are described.

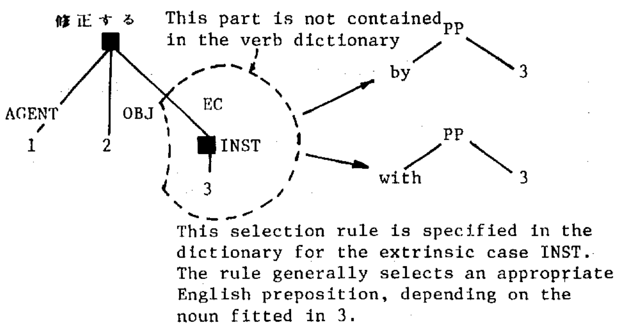



Fig. 12 JIS-EIS mapping for Extrinsic Cases

## 3-2. English Intermediate Structure — EIS

The EIS's are similar to conventional phrase structures. The main difference is that; each node in the tree is characterized not only by a category symbol like S, NP, VP, etc., but also by a set of attribute - value pairs. EIS plays almost the same role of 'starting phrase structure' in Chomsky. Successive transformations are applied cyclically on this structure during the English synthesis. However, the transformation component in our system includes a set of rules which are not 'structure dependent' and, therefore, not considered as 'transformation' in TG's sense. For example, passivized constructions are generated not through transformations in Chomsky's current framework, but they are considered as base-generated. In our system, however, they should be treated during English synthesis phase, whether they are structure dependent or not. The main purpose of transformations in the English synthesis is to generate adequate English surface structures from 'Japanese-generated' structures, instead of 'base-generated' ones. Passivization transformation, for example, is indispensable in our system, because it is common in Japanese to state sentences in active voice without any agents. In order to support such transformations, information other than syntactic categories and structures is necessary. They are expressed in EIS's as a set of attribute-value pairs attached to a node.

## 3-3. The Transfer Procedure

The general algorithm for the transfer phase changes a given JIS into the corresponding EIS by 'evaluating' the nodes in the JIS recursively.

Each JIS node is labelled by an intermediate word of Japanese which has a unique entry in the bi-lingual dictionary. The description in the dictionary contains a set of transfer procedures which show how to transfer the JIS substructures whose roots are the entry word. Each transfer procedure may be accompanied with a set of preconditions, if necessary. These preconditions are expressed by user defined LISP functions to examine the surrounding JIS as to whether the transfer procedure is appropriate or not. Some built-in LISP functions are provided to facilitate encoding these preconditions. If a JIS word has several English equivalents (i.e. it is polysemy relative to English), these preconditions are used to choose an appropriate one. Though deep semantic checking should be performed in this precondition part in more advanced systems, this part is currently used to examine certain syntactic environments or simple semantic markers.

A transfer procedure usually works as follows:
(1) A transfer procedure defined for a governing word (verb, relation-descriptor, etc.) will invoke the main program in order to transfer the JIS substructures governed by the current node.
(2) When these substructure transfers are completed, the transfer procedure attached to the governing node will arrange the substructures (in EIS) into single structures and return them to the higher level. Because transfer procedures

at the lower level generally return several possible EIS structures, the procedure at the higher level selects feasible combinations and returns them in parallel, if several combinations are feasible.
(3) A transfer procedure for a dependent word (typically noun) will not invoke the main program, but only choose the appropriate English equivalents. So the recursive process terminates.

Notice that the whole process is highly lexicon driven. Because the main program only checks the preconditions and invokes transfer procedures defined in the dictionary, we can easily change and augment the transfer step by adding new descriptions in the dictionary. Several standard transfer procedures are provided as shown in Table 3. Because these standard procedures are parameterized, most of Japanese intermediate words can be defined by supplying them with appropriate parameters. Fig. 13 shows an example of a verb dictionary which uses the standard procedure VB1 (specified in PNAME). VB1 transfers an input JIS to the EIS as shown in Fig. 13. Moreover, whenever we recognize that a certain intermediate word requires a special treatment, we can tailor a transfer procedure applied only for that word, and put it in the dictionary. This gives us a flexible framework for dealing exceptional words that cannot be managed by general procedures.

| Procedure | Generated EIS | Procedure | Generated EIS |
|---|---|---|---|
| VB-1<br>VB-2<br>: | The structures governed by verbs | COM-N<br>NOM-1 | Common nouns<br>Nominalized forms for sentences |
| REL-1 | Type I relative clause | QUANT-NOM | quantified noun phrases |
| REL-2 | Type II relative (whose type) cl. | LOC-1<br>: | prepositional phrases for extrinsic cases |
| REL-3 | Type III relative (THAT- COMP) cl. | TIME-1<br>: | |
| CONJ-1 | Conjunctive const. for co-ordinate clauses and nouns | COMP-N<br>IN-ORDER | Sentences with sentential comp.<br>The infinitive cl. 'in order to' |
| CONJ-2 | Conjunctive const. for subordinate clauses | TOUGH | Noun phrases with 'TOUGH' adjectives |

Table 3 Standard Transfer Procedures Used in the Bi-lingual Dictrionary

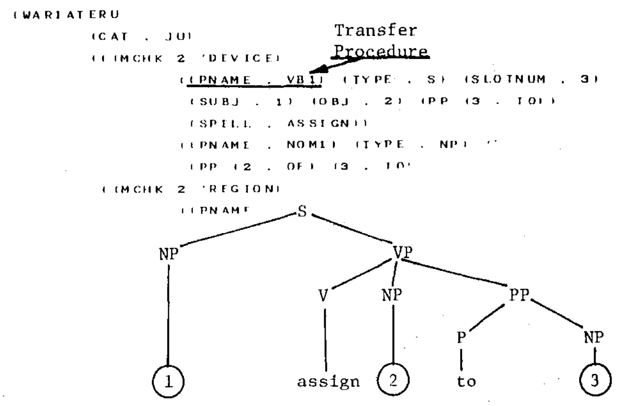
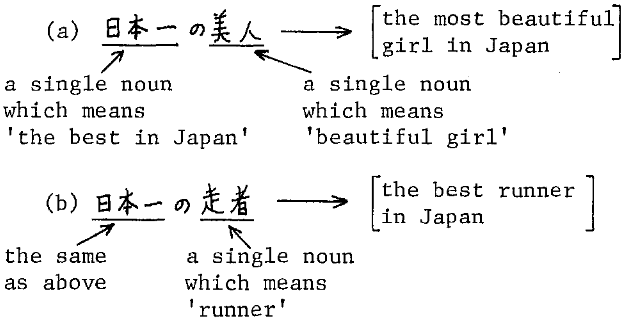



Fig. 13 Bi-lingual Dictionary for 'WARIATERU' (To Assign) and Its Parametorized EIS

We will pick up an example to illustrate this point.

The Japanese compound word '日本一' roughly means 'the best in Japan', and consists of two words, 日本 (Japan) and 一 (the first or one). Because the word behaves syntactically as a noun, the analysis procedure treats it as a usual noun. As usual nouns in Japanese, it can be used as a noun modifier.




The above two phrases are simply represented in JIS's as shown in Fig. 14. However, these phrases should be paraphrased in English. A special procedure is tailored and put in the lexicon for such a kind of words like 日本一 (the best in Japan), 世界一 (the best in the world) etc.

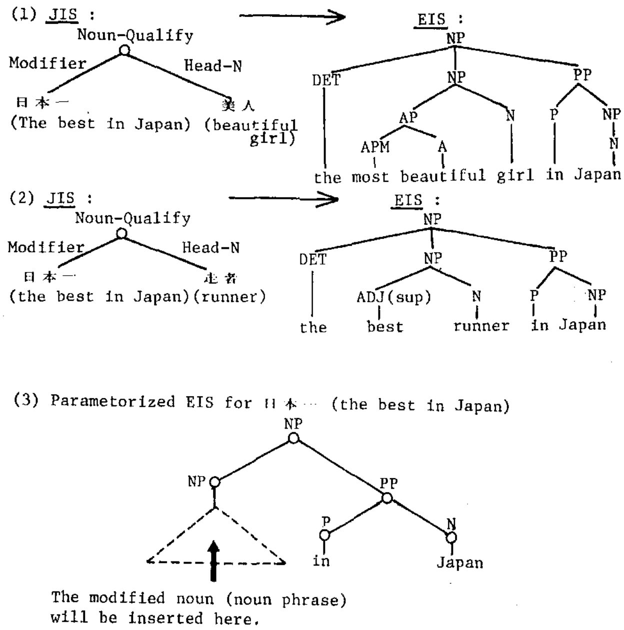



Fig. 14 Structural Transfer for the Noun 日本一 (the Best in Japan)

The procedure works as follows:

1. It checks whether the modified noun (or noun phrase) contains an adjective or not.
2. If it contains, the procedure attaches the superlative indicator to the adjective.
3. If it does not, the procedure supplies to the noun the default adjective 'good' with the superlative indicator.

4. It embeds the modified noun (or noun phrase) in the parametorized EIS structure as shown in Fig. 14-(3).

Notice that both the superlative transformation and the 'the' attachment to the superlative adjective will be done at the last step of the English synthesis phase.

## 4. English Synthesis

Because an EIS is generated directly from the corresponding JIS, it preserves many characteristics of Japanese syntax. In this sense, it is 'Japanese-generated' but not 'base-generated'. We should transform this structure to obtain a correct English syntactic structure. Japanese 'wh'-questions, for example, are stated in the forms similar to their declarative ones, except that wh-words are marked by special prefix words. The wh-movement rule is undoubtedly necessary to produce correct English sentences. Moreover, though passivization is not considered as a transformation from Lexicalists' point of view, it is indispensable in our system. Therefore, much information other than structural matching is necessary to determine whether the transformation rule is applicable or not.

### 4-1. The Generation Dictionary

At the first step of the generation, the system retrieves the lexical description of each word in the EIS from the generation dictionary. The generation dictionary contains information such as shown in Table 4. It contains not only trivial indicators necessary for morphological synthesis, but also some other indicators which are examined during the transformation process.

| marker | meaning | marker | meaning |
|---|---|---|---|
| UN-PASSIVE | Verbs which can not be used in passive | S-AFT | S-ADV adverbs which usually appears at the end of sentences |
| STATE | Verbs whose aspectual feature are 'STATE' | UNCOUNT | Uncountable nouns |
| V-ADV | Adverbs mostly used as verb modifiers | TOUGH | Tough adjectives |
| | | PROPER | Proper nouns |
| S-ADV | Adverbs mostly used as sentence modifiers | AN | The words that begin with vowels |
| VP-TOP | Adverbs usually preceding the verbs | S*S | The last characters of the words are 'ses',etc |
| S-TOP | S-ADV adverbs which usually appears at the beginning of sent. | INF | The words which has irregular inflection forms |

Table 4 Markers in the Synthesis Dictionary

### 4-2. Transformation Rule

A transformation rule is represented in our system by a 9-tuple as shown in Fig. 15. A transformation rule is essentially a tree-to-tree mapping expressed by MP ->CP. Each rule is specified as either OB or OP. OB means that the rule is obligatory; if the rule is applicable, it should be applied. If a rule is marked as OP(tional), it may or may not be applied.

At present, when an applicable optional rule is encountered, two alternative structures with equal feasibilities will be generated. To select

(NAME COM TYPE MP BPL RP PL IAL INAL)

NAME : The name of the rule.
COM : Comment. This does not have any actual effects. Only for later references and debuggings.
TYPE : This indicates whether the rule is obligatory (OB) or optional (OP).
MP : Matching Pattern which shows the tree schema on which the rule is to be applied.
BPL : Procedural descriptions for checking the applicability of the rule.
RP : Resultant pattern which shows the transformed tree structure.
IAL : If-applied list. This list contains the names of the rules that are to be applied if this rule is successfully applied.
INAL : If-not-applied list. This list contains the names of the rules which are to be applied if this rule fails.
PL : Program List which contains the programs which are applied to the transformed structure after the rule application succeeds.

Fig. 15 Format of a Transformation Rule

the most appropriate one would require certain stylistic considerations, which is beyond our current scope.

The applicability of a rule is checked not only by pattern-matching but also by user-defined checking procedures specified in BPL. Because an MP contains several variables and the pattern-matching between MP and the current tree structural binds the variables to appropriate substructures, these user-defined procedures can investigate the relationships between substructures in arbitrary ways, including attribute checkings, by utilizing this variable binding.

The whole algorithm works cyclically from bottom to top, as usual transformations. According to the rule map as illustrated in Fig. 16, transformation rules are applied to every cyclic node (VP, NP, S) at the lowest in a tree, then at one level higher, and so on.

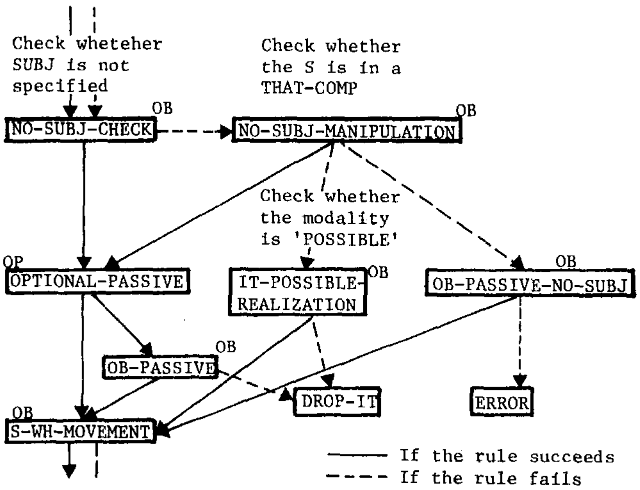



Fig. 16 . Rule Map for English Synthesis

The system currently has about 200 rules which are selected from (7). After the major transformation cycle is finished, English morphological synthesis will begin which traverses the resultant tree structures to generate appropriate morphological variants. No special comments would be necessary for this phase.

## 5. Concluding Remarks

Fig. 17. shows some examples of translation which illustrate the current abilities of the system. As these examples show, the system can translate fairly complex sentences, though several problems still remain unsolved. The distinction between definite and indefinite noun phrases, for example, cannot be made by the current system, because no fixed expressions to distinguish them exist in Japanese surface sentences. Therefore, neither definite nor indefinite articles are not attached to the English noun phrases. Another problem is to supply appropriate elements from context for omitted expressions. Especially, case elements in a sentence are frequently omitted in Japanese, when they are easy to recognize from the context. Though the current system tries to find appropriate surface English words and structures at the English synthesis phase which do not require the omitted elements, it would be inevitable to incorporate contextual processings. The current system works very well as an experimental prototype. Following to the same basic principles with the current system, we are now designing a new and more advanced system, in which these defects of the current system will be improved.

Our basic contention in this paper is that most of linguistic phenomena should be treated by lexical rules, instead of general syntactic rules. This leads us to the framework called lexicon based procedures. This approach is not only fairly compatible with the recent trends in linguistics, but also gives us a good framework in which grammars can be easily revised and augmented by modifying the lexical description of each individual word, without any modifications of the general framework.

The next comment is about the relationships among syntactic, semantic, and pragmatic processings. At the early stage of development, the researchers of natural language understanding systems, including the authors, emphasized too much the importance of pragmatic knowledge. However, one of the recent trends in this area, which we also support, is to lay more emphasis on the importance of syntactic processings, or at least, syntactic structures of sentences. This attitude is, we believe, especially important for MT systems. The various transformed syntactic structures described in section 2 have been overlooked by the researchers of computational linguistics so far. We hope that our approach, the lexicon based analysis procedure, provides an appropriate framework to integrate syntactic structures and operations with the other kinds of processings such as semantic and pragmatic ones.

## References


(1)W.Hutchins : Machine Translation and Machine-Aided Translation, Jour. of Doc., 1978

(2)J.Bresnan : Realistic Transformational Grammar, in Linguistic Theory and Psychological Reality, MIT Press, 1979

(3)N.Chomsky : On Wh-movement, in Formal Syntax, Academic Press, 1977

(4)M.Nagao et.al.: Analysis of Japanese Sentences by Using Sementic and Contextual Information, AJCL, Microfiche 41, 1976

(5)K.Mitamura : Japanese Analysis for a MT system, MS Thesis, Kyoto Univ., 1980 (in Japanese)

(6)C.Rieger : Word Expert Parsing, in Proc. of the 6 th IJCAI, Tokyo, 1979

(7)R.Stockwell et.al. : The Major Syntactic Structures of English, Holt,Rinehart,& Winston,1973


(1) Input Japanese 順データセットに対しては、ユーザ見出しラベル及びユーザ終わりラベルを書き出すことができる。

Result of Translation

#1 ***

User header label and user trailer label can be written out to sequential data set.

#2 ***

It is possible to write out user header label and user trailer label to sequential data set.

(2) Input Japanese ＤＤ文のＤＳＮＡＭＥパラメタにメンバ名を指定した場合、ユティリティ制御文にＮＡＭＥパラメタが指定してあればＮＡＭＥパラメタによって指定されたメンバ名が優先する。

Result of Translation

#1 ***

When member name was specified in DSNAME parameter of DD statement, if NAME parameter is specified in utility control statement, member name specified by NAME parameter takes precedance.

#2 ***

When member name was specified in DSNAME parameter of DD statement, if NAME parameter is specified in utility control statement, member name which is specified by NAME parameter takes precedance.

Fig. 17. Translation Results