A PROPER TREATMEMT OF SYNTAX AND SEMANTICS IN MACHINE TRANSLATION

Yoshihiko Nitta, Atsushi Okajima, Hiroyuki Kaji,
Youichi Hidano, Koichiro Ishihara

Systems Development Laboratory, Hitachi, Ltd.
1099 Ohzenji Asao-ku, Kawasaki-shi, 215 JAPAN

ABSTRACT

A proper treatment of syntax and semantics in machine translation is introduced and discussed from the empirical viewpoint. For English-Japanese machine translation, the syntax directed approach is effective where the Heuristic Parsing Model (HPM) and the Syntactic Role System play important roles. For Japanese-English translation, the semantics directed approach is powerful where the Conceptual Dependency Diagram (CDD) and the Augmented Case Marker System (which is a kind of Semantic Role System) play essential roles. Some examples of the difference between Japanese sentence structure and English sentence structure, which is vital to machine translation, are also discussed together with various interesting ambiguities.

I INTRODUCTION

We have been studying machine translation between Japanese and English for several years. Experiences gained in systems development and in linguistic data investigation suggest that the essential point in constructing a practical machine translation system is in the appropriate blending of syntax directed processing and the semantics directed processing.

In order to clarify the above-mentioned suggestion, let us compare the characteristics of the syntax directed approach with those of the semantics directed approach.

The advantages of the syntax directed approach are as follows:

(1) It is not so difficult to construct the necessary linguistic data for syntax directed processors because the majority of these data can be reconstructed from already established and well-structured lexical items such as verb pattern codes and parts of speech codes, which are overflowingly abundant in popular lexicons.

(2) The total number of grammatical rules necessary for syntactic processing usually stays within a controllable range.

(3) The essential aspects of syntactic processing are already well-known, apart from efficiency problems.

The disadvantage of the syntax directed approach is its insufficient ability to resolve various ambiguities inherent in natural languages.

On the other hand, the advantages of the semantics directed approach are as follows:

(1) The meaning of sentences or texts can be grasped in a unified form without being affected by the syntactic variety.

(2) Semantic representation can play a pivotal role for language transformation and can provide a basis for constructing a transparent machine translation system, because semantic representation is fairly independent of the differences in language classes.

(3) Consequently, semantics directed internal representation can produce accurate translations.

The disadvantages of the semantics directed approach are as follows:

(1) It is not easy to construct a semantic lexicon which covers real world phenomena of a reasonably wide range. The main reason for this difficulty is that a well-established and widely-accepted method of describing semantics does not exist. (For strongly restricted statements or topics, of course, there exist well-elaborated methods such as Montague grammar [2], Script and MOP (Memory Organization Packet) theory [13], Procedural Semantics [14], and Semantic Interlingual Representation [15].)

(2) The second but intractable problem is that, even if you could devise a fairly acceptable method to describe semantics, the total number of semantic rule descriptions may expand beyond all manageable limits.

Therefore, we think that it is necessary to seek proper combinations of syntactic processing and semantic processing so as to compensate for the disadvantages of each.

The purpose of this paper is to propose a proper treatment of syntax and semantics in machine translation systems from a heuristic viewpoint, together with persuasive examples obtained through operating experiences. A sub-language approach which would put some moderate restrictions on the syntax and semantics of source language is also discussed.

II SYNTAX AND SEMANTICS

It is not entirely possible to distinguish a syntax directed approach from a semantics directed approach, because syntax and semantics are always performing their linguistic functions reciprocally.

As Wilks [16] points out, it is plausible but a great mistake to identify syntactic processing with superficial processing, or to identify semantic processing with deep processing. The term "superficial" or "deep" only reflects the intuitive distance from the language representation in (superficial) character strings or from the language representation in our (deep) minds. Needless to say, machine translation inevitably has something to do with superficial processing.

In various aspects of natural language processing, it is quite common to segment a superficial sentence into a collection of phrases. A phrase itself is a collection of words. In order to restructure the collection of phrases, the processor must first of all attach some sorts of labels to the phrases. If these labels are something like subject, object, complement, etc., then we will call this processor a syntax directed processor, and if these labels are something like agent, object, instrument, etc., or animate, inanimate, concrete, abstract, human, etc., then we will call this processor a semantics directed processor.

The above definition is oversimplified and of course incomplete, but it is still enough for the arguments in this paper.

III SYNTAX DIRECTED APPROACH: A PROTOTYPE ENGLISH-JAPANESE MACHINE TRANSLATION SYSTEM

So far we have developed two prototype machine translation systems; one is for English-Japanese translation [6] and the other is for Japanese-English translation.

The prototype model system for English-Japanese translation (Figure 1) is constructed as a syntax directed processor using a phrase structure type internal representation called HPM (Heuristic Parsing Model), where the semantics is utilized to disambiguate dependency relationships.

The somewhat new name HPM (Heuristic Parsing Model) reflects the parsing strategy by which the machine translation tries to simultate the heuristic way of actual human of language translation. The essential features of heuristic translation are summarized in the following three steps:

(1) To segment an input sentence into phrasal elements (PE) and clausal elements (CE).

(2) To assign syntactic roles to PE's and CE's, and restructure the segmented elements into tree-forms by governing relation, and into link-forms by modifying relation.

(3) To permute the segmented elements, and to assign appropriate Japanese equivalents with necessary case suffixes and postpositions.

Noteworthy findings from operational experience and efforts to improve the prototype model are as follows:

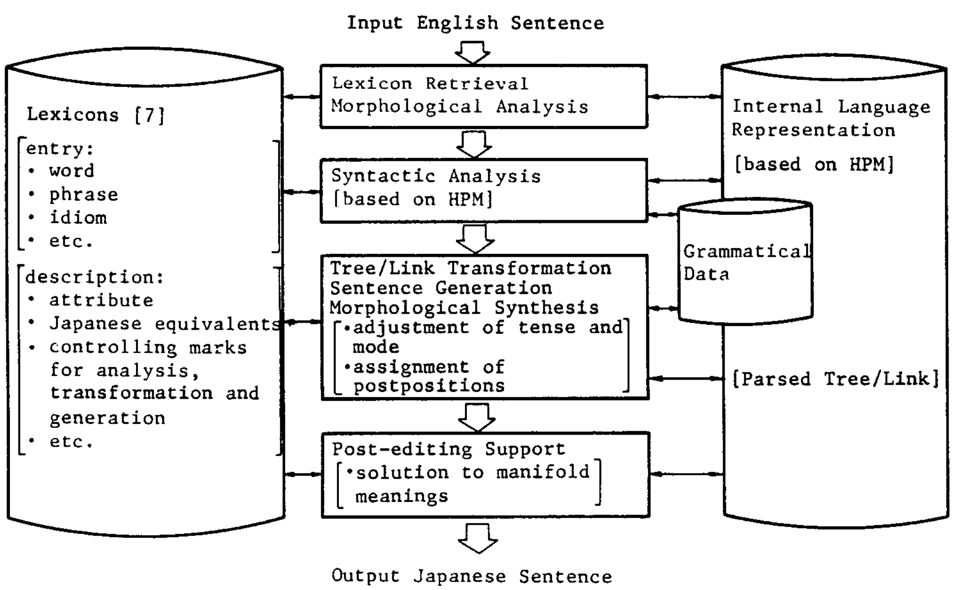
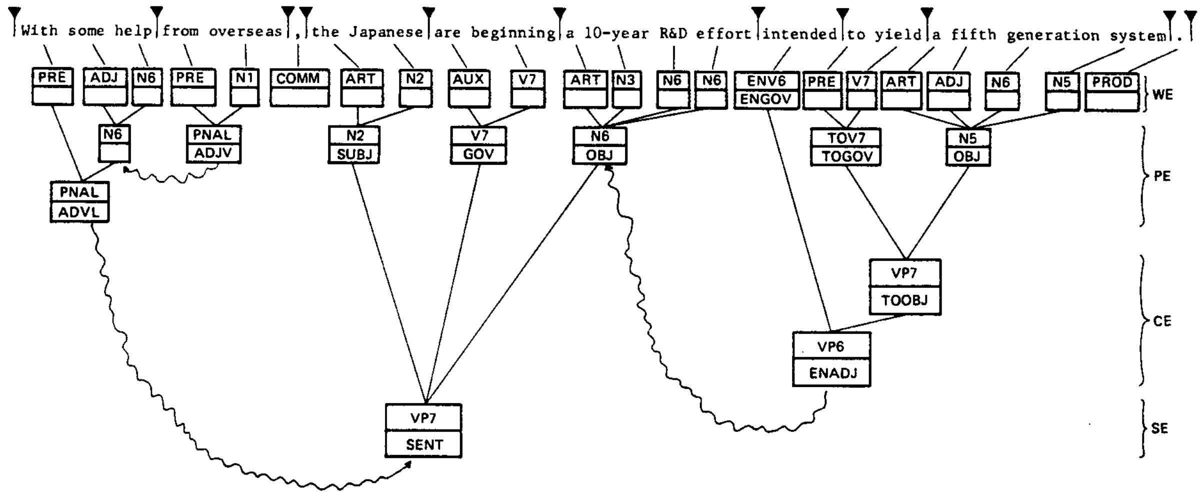


Figure 1 Configuration of Machine Translation System: ATHENE [6]



• WE: Word Element
• PE: Phrasal Element
• CP: Clausal Element
• SE: Sentence
• This sample English sentence is taken from Datamation Jan. 1982.

Figure 2 An Example of Phrase Structure Type Representation

(1) The essential structure of English sentences should be grasped by phrase structure type representations.

An example of phrase strucure type representation, which we call HPM (Heuristic Parsing Model), is illustrated in Figure 2. In Figure 2, a parsed tree is composed of two substructures. One is "tree (\|/)," representing a compulsory dependency relation, and the other is "link (∿)," representing an optional dependency relation. Each node corresponds to a certain constituent of the sentence.

The most important constituent is a "phrasal element (PE)" which is composed of one or more word element(s) and carries a part of the sentential meaning in the smallest possible form. PE's are mutually exclusive. In Figure 2, PE's are shown by using the "segmenting marker (⊺)", such as

⊺With some help (ADVL)⊺,
⊺from overseas (ADJV)⊺,
⊺,(COMM)⊺,
⊺the Japanese (SUBJ)⊺
and
⊺are beginning (GOV)⊺,

where the terminologies in parentheses are the syntactic roles which will be discussed later.

A "clausal element (CE)" is composed of one or more PE('s) which carries a part of sentential meaning in a nexus-like form. A CE roughly corresponds to a Japanese simple sentence such as: "∿{wa/ga/wo/no/ni} ∿ {suru/dearu} [koto]." CE's allow mutual intersection. Typical examples are the underlined parts in the following:

"It is important for you to do so."

"... intended to yield a fifth generation system."

One interesting example in Figure 2 may be the part

"With some help from overseas",

which is treated as only two consecutive phrasal elements. This is the typical result of a syntax directed parser. In the case of a semantics directed parser, the above-mentioned part will be treated as a clausal element. This is because the meaning of this part is "(by) getting some help from overseas" or the like, which is rather clausal than phrasal.

(2) Syntax directed processors are effective and powerful to get phrase structure type parsed trees.

Our HPM parser operates both in a top-down way globally and in a bottom-up way locally. An example of top-down operation would be the segmentation of an input sentence (i.e. the sequence of word elements (WE's)) to get phrasal elements (PE), and an example of bottom-up operation would be the construction of tree-forms or link-forms to get clausal elements (CE) or a sentence (SE). These operations are supported by syntax directed grammatical data such as verb dependency type codes (cf. Table 1, which is a simplified version of Hornby's classification [5]), syntactic role codes (Table 2) and some production rule type grammars (Table 3 & Table 4). It may be permissible to say that all these syntactic data are fairly compact and the kernel parts are already well-elaborated (cf. [1], [8], [11], [12]).

Table 1 Dependency Pattern of Verb

Code	Verb Pattern	Examples
V1	Be + ...	be
V2	Vi (≠ Be) + Complement, It/There + Vi + ...	get, look
V3	Vi [+ Adverbial Modifier]	rise, walk
V6	Vt + To-infinitive	intend
V7	Vt + Object	begin, yield
V8	Vt + that + ...	agree, think
V14	Vt + Object [+not] + To-infinitive	know, bring

Table 2 Syntactic Roles

Code	Role
SUBJ	Subject
OBJ	Object
TOOBJ	Object in To-infinitive Form
NAPP	Noun in Apposition
GOV	Governing Verb
TOGOV	Governing Verb in To-infinitive Form
ENGOV	Governing Verb in Past Participle Form
ADJV	Adjectival
ENADJ	Adjectival in Past Participle Form
ADVL	Adverbial
SENT	Sentence

(3) The weak point of syntax directed processors is their insufficient ability to disambiguate; i.e. the ability to identify dependency types of verb phrases and the ability to determine heads of prepositional phrase modifiers.

(4) In order to boost the aforementioned disambiguation power, it is useful to apply semantic filters that facilitate the selective restrictions on linking a verb with nominals and on linking a modifier with its head.

A typical example of the semantic filter is illustrated in Figure 3. The semantic filter may operate along with selective restriction rules such as:

- N_{22} (Animal) + with + N_{753} (Accessory) ⟶ Plausible
  [∵ N_{22} is equipped with N_{753}]
- V_{21} (Watching-Action) + with + N_{541} (Watching Instrument) ⟶ OK
  [∵ V_{21} by using N_{541} as an instrument]

The semantic filter is not complete, especially for metaphorical expressions. A bird could also use binoculars.

Table 3 Rules for Assigning Syntactic Roles to Phrasal Elements

	Pattern to be Scanned	New Pattern to be Generated
1	$\begin{Bmatrix}\text{PRE}\\ \text{—}\end{Bmatrix} + \begin{Bmatrix}\text{N6}\\ \text{—}\end{Bmatrix} + \begin{Bmatrix}\text{PNAL}\\ \phi\end{Bmatrix}^{*} + \begin{Bmatrix}\text{COMM}\\ \text{—}\end{Bmatrix}$	$\begin{Bmatrix}\text{PRE}\\ \text{—}\end{Bmatrix} + \begin{Bmatrix}\text{N6}\\ \text{—}\end{Bmatrix} + \begin{Bmatrix}\text{PNAL}\\ \text{ADJV}\end{Bmatrix}^{*} + \begin{Bmatrix}\text{COMM}\\ \text{—}\end{Bmatrix}$
2	$\begin{bmatrix}\text{COMM}\\ \text{—}\end{bmatrix} + \begin{Bmatrix}\text{N}\\ \phi\end{Bmatrix} + \begin{Bmatrix}\text{V7}\\ \phi\end{Bmatrix}^{*} + \begin{Bmatrix}\text{N}\\ \phi\end{Bmatrix}$	$\begin{bmatrix}\text{COMM}\\ \text{—}\end{bmatrix} + \begin{Bmatrix}\text{N}\\ \text{SUBJ}\end{Bmatrix} + \begin{Bmatrix}\text{V7}\\ \text{GOV}\end{Bmatrix}^{*} + \begin{Bmatrix}\text{N}\\ \text{OBJ}\end{Bmatrix}$
3	$\begin{Bmatrix}\text{TOV7}\\ \phi\end{Bmatrix}^{*} + \begin{Bmatrix}\text{N}\\ \phi\end{Bmatrix}$	$\begin{Bmatrix}\text{TOV7}\\ \text{TOGOV}\end{Bmatrix}^{*} + \begin{Bmatrix}\text{N}\\ \text{OBJ}\end{Bmatrix}$
4	$\begin{Bmatrix}\text{ENV}\\ \phi\end{Bmatrix}^{*} + \begin{Bmatrix}\text{TOV}\\ \text{—}\end{Bmatrix}$	$\begin{Bmatrix}\text{ENV}\\ \text{ENGOV}\end{Bmatrix}^{*} + \begin{Bmatrix}\text{TOV}\\ \text{—}\end{Bmatrix}$
5	$\phi + \begin{Bmatrix}\text{PRE}\\ \phi\end{Bmatrix}^{*} + \begin{Bmatrix}\text{N}\\ \text{–}\end{Bmatrix}^{*}\left[+\begin{Bmatrix}\text{—}\\ \text{ADJV}\end{Bmatrix}\right] + \begin{Bmatrix}\text{—}\\ \text{SENT}\end{Bmatrix}$	$\phi + \begin{Bmatrix}\text{PNAL}\\ \text{ADVL}\end{Bmatrix}^{*} + \begin{Bmatrix}\text{—}\\ \text{SENT}\end{Bmatrix}$

*: focus, —: not mentioned, ϕ: empty, [...]: optional

Table 4 Rules for Constructing Clausal Elements

	Pattern to be Scanned	New Element to be Generated
1	$\begin{Bmatrix}\text{—}\\ \text{SUBJ}\end{Bmatrix} + \begin{Bmatrix}\text{V7}\\ \text{GOV}\end{Bmatrix}^{*} + \begin{Bmatrix}\text{—}\\ \text{OBJ}\end{Bmatrix}$	$\begin{Bmatrix}\text{VP7}\\ \text{SENT}\end{Bmatrix}^{*}$
2	$\begin{Bmatrix}\text{[EN]V6}\\ \text{—}\end{Bmatrix} + \begin{Bmatrix}\text{—}\\ \text{TOGOV}\end{Bmatrix}^{*} + \begin{Bmatrix}\text{—}\\ \text{OBJ}\end{Bmatrix}^{*}$	$\begin{Bmatrix}\text{[EN]V6}\\ \text{—}\end{Bmatrix} + \begin{Bmatrix}\text{—}\\ \text{TOOBJ}\end{Bmatrix}^{*}$
3	$\begin{Bmatrix}\text{N}\\ \text{–}\end{Bmatrix} + \begin{Bmatrix}\text{—}\\ \text{ENGOV}\end{Bmatrix}^{*} + \begin{Bmatrix}\text{—}\\ \text{[TO]OBJ}\end{Bmatrix}^{*}$	$\begin{Bmatrix}\text{N}\\ \text{–}\end{Bmatrix} + \begin{Bmatrix}\text{—}\\ \text{ENADJ}\end{Bmatrix}$

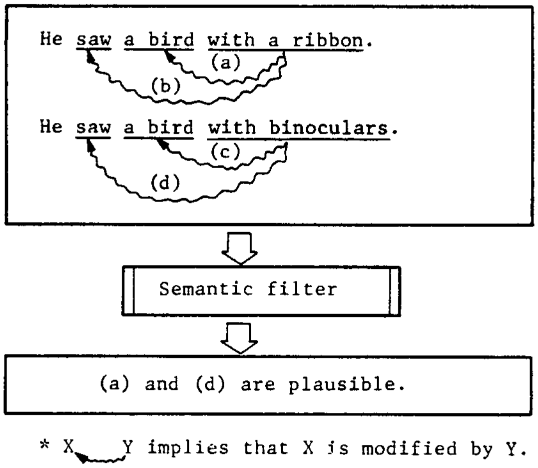


Figure 3 A Typical Operation of Semantic Filter

(5) The aforementioned semantic filters are compatible with syntax directed processors; i.e. there is no need to reconstruct processors or to modify internal representations. It is only necessary to add filtrating programs to the syntax directed processor.

One noteworthy point is that the thesaurus for controlling the semantic fields or semantic features of words should be constructed in an appropriate form (such as word hierarchy) so as to avoid the so-called combinatorial explosion of the number of selective restriction rules.

(6) For the Japaneses sentence generating process, it may be necessary to devise a very complicated semantic processor if a system to produce natural idiomatic Japanese sentences is desired. But the majority of Japanese users may tolerate awkward word-by-word translation and understand its meaning. Thus we have concluded that our research efforts should give priority to the syntax directed analysis of English sentences. The semantics directed generation of Japanese sentences might not be an urgent issue; rather it should be treated as a kind of profound basic science to be studied without haste.

(7) Even though the output Japanese translation may be an awkward word-by-word translation, it should be composed of pertinent function words and proper equivalents for content words. Otherwise it could not express the proper meaning of the input English sentences.

(8) In order to select proper equivalents, semantic filters can be applied fairly effectively to test the agreement among the semantic codes assigned to words (or phrases).

Again the semantic filter is not always complete. For example, in Figure 2, the verb "yield" has at least two different meanings (and consequently has at least two different Japanese equivalents):

"yield" → { "produce" (= Umidasu)
"concede" (= Yuzuru).

But it is neither easy nor certain how to devise a filter to distinguish the above two meanings mechanically. Thus we need some human aids such as post-editing and inter-editing.

(9) As for the pertinent selection of function words such as postpositions, there are no formal computational rules to perform it. So we must find and store heuristic rules empirically and then make proper use of them.

Some heruistic rules to select appropriate Japanese postpositions are shown in Table 5.

Table 5 Heuristic Rules for Selecting Postpositions for "in + N"

Semantic Category of N	Japanese Post-positions for ADVL/ADJV	English Examples
in+N1 (N1=Place)	N1+de/N1+niokeru	in California
in+N3 (N3=Time)	N3+ni/N3+no	in Spring
in+N3&N4 (N4=Quantity)	—/N3&N4+go-ni	in two days
in+N6 (N6=Abstract Concept)	N6+dewa/N6+no	in my opinion
in+N8 (N8=Means)	N8+de/N8+niyoru	in Z-method
. No rules.	+de/+no	(speak) in English
. A kind of idiom [7] to be retrieved directly from a lexicon.	+wo-kite/ +wo-kita	in uniform
	+wo-kakete/ +wo-kaketa	in spectacles

(10) To get back to the previous findings (1) and (2), the heuristic approach was also found to be effective in segmenting the input English sentence into a sequence of phrasal elements, and in structuring them into a tree-like dependency diagram (cf. Figure 2).

(11) A practical machine translation should be considered from a kind of heuristic viewpoint rather than from a purely rigid analytical linguistic viewpoint. One persuasive reason for this is the fact that humans, even foreign language learners, can translate fairly difficult English sentences without going into the details of parsing problems.

IV SEMANTICS DIRECTED APPROACH: A PROTOTYPE JAPANESE-ENGLISH MACHINE TRANSLATION SYSTEM

The prototype model system for Japanese-English translation is constructed as a semantics directed processor using a conceptual dependency diagram as the internal representation. Noteworthy findings through operational experience and efforts to improve on the prototype model are as follows:

(1) Considering some of the characteristics of the Japanese language, such as flexible word ordering and ambiguous usage of function words, it is not advantageous to adopt a syntax directed representation for the internal base of language transformation.

For example, the following five Japanese sentences have almost the same meaning except for word ordering and a subtle nuance. Lowercase letters represent function words.

Boku wa Fude de Tegami wo Kaku.
(I) (brush)(with)(letter) (write)

Boku wa tegami wo Fude de Kaku.
Fude de Boku wa Tegami wo Kaku.

Tegami wa Boku wa Fude de Kaku.
Boku wa Tegami wa Fude de Kaku.

(2) Therefore we have decided to adopt the conceptual dependency diagram (CDD) as a compact and powerful semantics directed internal representation.

Our idea of the CDD is similar to the well-known dependency grammar defined by Hays [4] and Robinson [9] [10], except for the augmented case markers which play essentially semantic roles.

(3) The conceptual dependency diagram for Japanese sentences is composed of predicate phrase nodes (PPNs in abbreviation) and nominal phrase nodes (NPNs in abbreviation). Each PPN governs a few NPNs as its dependants. Even among PPNs there exist some governor-dependant relationships.

Examples of formal CDD description are:

PPN (NPN1, NPN2, ... NPNn),
Kaku (Boku, Tegami, Fude),
Write (I, Letter, Brush),

where the underlined word "α" represents the concept code corresponding to the superficial word "α", and the augmented case markers are omitted.

In the avove description, the order of dependants N1, N2, ..., Nn are to be neglected. For example,

PPN (NPNn, ..., NPN2, NPN1)

is identical to the above first formula. This convention may be different from the one defined by Hays [4]. Our convention was introduced to cope with the above-mentioned flexible word ordering in Japanese sentences.

(4) The aforementioned dependency relationships can be represented as a linking topology, where each link has one governor node and one dependant node as its top and bottom terminal point (Figure 4).

(5) The links are labeled with case markers. Our case marker system is obtained by augmenting the traditional case markers such as Fillmore's [3] from the standpoint of machine translation. For the PPN-NPN link, its label usually represents agent, object, goal, location, topic, etc. For the PPN-PPN link, its label is usually represent causality, temporality, restrictiveness, etc. (cf. Figure 4).

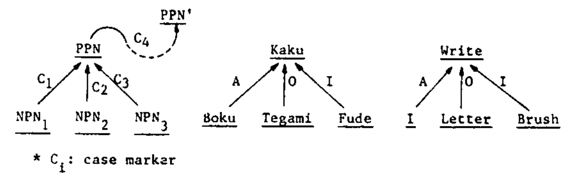


Figure 4 Examples of a Conceptual Dependency Diagram (CDD)

(6) As for the total number of case markers, our current conclusion is that the number of compulsory case markers to represent predicative dominance should be small, say around 20; and that the number of optional case markers to represent adjective or adverbial modification should be large, say from 50 to 70 (Table 6).

(7) The reason for the large number of optional case markers is that the detailed classification of optional cases is very useful for making an appropriate selection of prepositions and participles (Table 7).

(8) Each NPN is to be labeled with some properly selected semantic features which are under the control of a thesaurus type lexicon. Semantic features are effective to disambiguate predicative dependency so as to produce an appropriate English verb phrase.

(9) The essential difference between a Japanese sentence and the equivalent English sentence can be grasped as the difference in the mode of PPN selections, taken from the viewpoint of conceptual dependency diagram (Figure 5). Once an appropriate PPN selection is made, it will be rather simple and mechanical to determine the rest of the dependency topology.

(10) Thus the essential task of Japanese-English translation can be reduced to the task of constructing the rules for transforming the dependency topology by changing PPNs, while preserving the meaning of the original dependency topology (cf. Figure 5).

(11) All the aforementioned findings have something to do with the semantic directed approach. Once the English oriented conceptual dependency diagram is obtained, the rest of the translation process is rather syntactic. That is, the phrase structure generation can easily be handled with somewhat traditional syntax directed processors.

(12) As is well known, the Japanese language has a very high degree of complexity and ambiguity mainly caused by frequent ellipsis and functional multiplicity, which creates serious obstacles for the achievement of a totally automatic treatment of "raw" Japanese sentences.

(ex 1) "Sakana wa Taberu."
(fish) (eat)
has at least two different interpretations:
. "[Sombody] can eat a fish."
. "The fish may eat [something]."

Table 6 Case Markers for CDD (subset only)

Predicative Dominance (Compulsory)	A	Agent
	O	Object
	C	Complement
	R	Recipient
	AC	Agent in Causative
	T	Theme, Topic (Mental Subject)
	P	Partner
	Q	Quote
	RI	Range of Interest
	RQ	Range of Qualification
	RM	Range of Mention
	I	Instrument
	E	Element
	...	...
Adverbial Modification (Optional)	CT	Goal in Abstract Collection
	CF	Source in Abstract Collection
	TP	Point in Time
	...	...
Adjective Modification (Optional)	ET	Embedding Sentence Type Modifier whose gapping is Theme
	EA	— whose gapping is Agent
	EO	— whose gapping is Object
	...	...
Link and Conjunction (Optional)	LA	Linking through "AND"
	BT	Conjunction through "BUT"
	...	...

(ex 2) "Kawaii Ningyou wo Motteiru Onnanoko."
(lovely) (doll) (carry) (girl)
has also two different interpretations:
. "The lovely girl who carries a doll with her."
. "The girl who carries a lovely doll with her."

(13) Thus we have judged that some sub-Japanese language should be constructed so as to restrict the input Japanese sentences within a range of clear tractable structures. The essential restrictions given by the sub-language should be concerned with the usage of function words and sentential embeddings.

Table 7 Detailed Classification of Optional Case Markers for Modification (subset only)

Phase Code	Most-Likely Prepositions or Participles
F	from
T	to, till
D	during
P	at
I	in, inside
O	out, outside
V	over, above
U	under, below
S	beside
B	before, in front of
A	after, behind
AL	along
H	through
AB	over, superior to
SE	apart from
WI	within
...	...

. Case Marker ≡ Body Code + Phase Code
. Body Code ≡ T (=Time)| S (=Space)| C (=Collection)

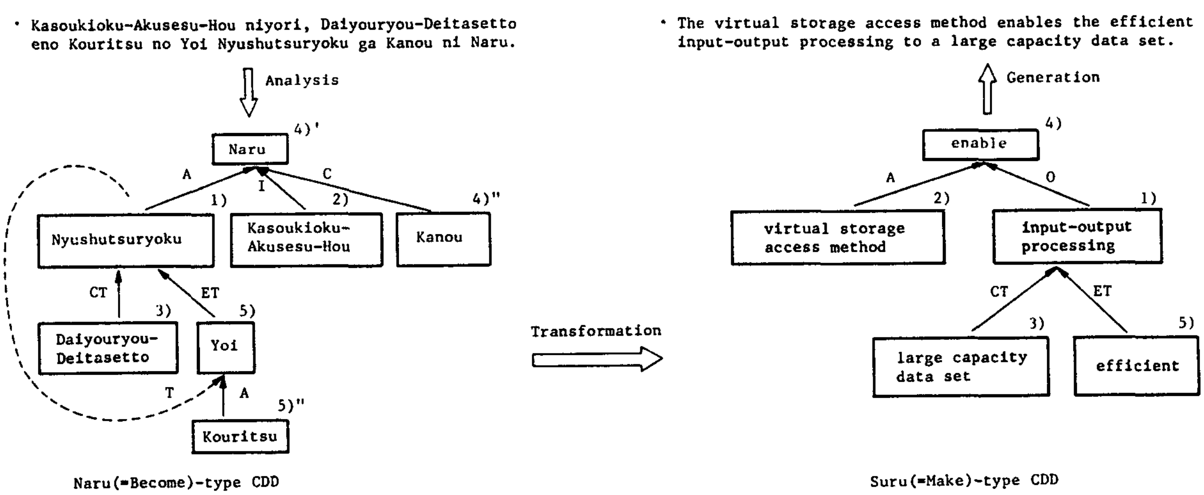


Figure 5 Difference between Japanese and English Grasped Through CDD

(14) A sub-language approach will not fetter the users, if a Japanese-English translation system is used as an English sentence composing aid for Japanese people.

V CONCLUSION

We have found that there are some proper approaches to the treatment of syntax and semantics from the viewpoint of machine translation. Our conclusions are as follows:

(1) In order to construct a practical English-Japanese machine translation system, it is advantageous to take the syntax directed approach, in which a syntactic role system plays a central role, together with phrase structure type internal representation (which we call HPM).

(2) In English-Japanese machine translation, syntax should be treated in a heuristic manner based on actual human translation methods. Semantics plays an assistant role in disambiguating the dependency among phrases.

(3) In English-Japanese machine translation, an output Japanese sentence can be obtained directly from the internal phrase structure representation (HPM) which is essentially a structured set of syntactic roles. Output sentences from the above are, of course, a kind of literal translation of stilted style, but no doubt they are understandable enough for practical use.

(4) In order to construct a practical Japanese-English machine translation system, it is advantageous to take the approach in which semantics plays a central role together with conceptual dependency type internal representation (which we call CDD).

(5) In Japanese-English machine translation, augmented case markers play a powerful semantic role.

(6) In Japanese-English machine translation, the essential part of language transformation between Japanese and English can be performed in terms of changing dependency diagrams (CDD) which involves predicate replacements.

One further problem concerns establishing a practical method of compensating a machine translation system for its mistakes or limitations caused by the intractable complexities inherent to natural languages. This problem may be solved through the concept of sublanguage, pre-editing and post-editing to modify source/target languages. The sub-Japanese language approach in particular seems to be effective for Japanese-English machine translaton. One of our current interests is in a proper treatment of syntax and semantics in the sublanguage approach.

ACKNOWLEDGEMENTS

We would like to thank Prof. M. Nagao of Kyoto University and Prof. H. Tanaka of Tokyo Institute of Technology, for their kind and stimulative discussion on various aspects of machine translation. Thanks are also due to Dr. J. Kawasaki, Dr. T. Mitsumaki and Dr. S. Mitsumori of SDL Hitachi Ltd. for their constant encouragement to this work, and Mr. F. Yamano and Mr. A. Hirai for their enthusiastic assistance in programming.

REFERENCES

[1] Chomsky, N., Aspects of the Theory of Syntax (MIT Press, Cambridge, MA, 1965).
[2] Dowty, D.R. et. al., Introduction to Montague Semantics (D. Reidel Publishing Company, Dordrecht: Holland, Boston: U.S.A., London: England, 1981)
[3] Fillmore, C.J., The Case for Case, in: Bach and Harms (eds.), Universals in Linguistic Theory, (Holt, Reinhart and Winston, 1968) 1-90
[4] Hays, D.G., Dependency Theory: A Formalism and Some Observations, Language, vol.40, no.4 (1964) 511-525
[5] Hornby, A.S., Guide to Patterns and Usage in English, second edition (Oxford University Press, London, 1975).
[6] Nitta, Y., Okajima, A. et. al., A Heuristic Approach to English-into-Japanese Machine Translation, COLING-82, Prague (1982) 283-288
[7] Okajima, A., Nitta, Y. et. al., Lexicon Structure for Machine Translation, ICTP-83, Tokyo (1983) 252-255
[8] Quirk et. al., A Grammar of Contemporary English (Longman, London; Seminar Press, New York, 1972).
[9] Robinson, J.J., Case, Category and Configuration, Journal of Linguistics, vol.6 no.1 (1970) 57-80
[10] Robinson, J.J., Dependency Structures and Transformational Rules, Language, vol.46, no.2 (1970) 259-285
[11] Robinson, J.J., DIAGRAM: A Grammar for Dialogues, Comm. ACM vol.25, no.1 (1982) 27-47.
[12] Sager, N., Natural Language Information Processing (Addison Wesley, Reading, MA., 1981).
[13] Schank, R.C., Reminding and Memory Organization: An Introduction to MOPs, in: Lehnert W.G. and Ringle, M.H. (eds.), Strategies for Natural Language Processing (Lawrence Erlbaum Associates, Publishers, Hillsdale, New Jersey, London, 1982) 455-493
[14] Wilks, Y., Some Thoughts on Procedural Semantics, in: ibid. 495-521
[15] Wilks, Y., An Artificial Intelligence Approach to Machine Translation, in: Schank, R.C. and Colby, K.M. (eds.), Computer Models of Thought and Language (W.H. Freeman and Company, San Francisco, 1973) 114-151
[16] Wilks, Y., Deep and Superficial Parsing, in: King, M. (ed.), Parsing Natural Language (Academic Press, London, 1983) 219-246