

MACHINE TRANSLATION OF THE URAL-ALTAIC AS AN AGGLUTINATIVE LANGUAGE

- ON THE KOREAN TO JAPANESE TRANSLATION -

Tae Wan Kim, Jin Tae Lee, Chang Ho Park, Ki Sik Lee

Systems Engineering Research Center
Korea Advanced Institute of Science & Technology
Seoul, Korea

ABSTRACT

In this paper we present the results of Korean to Japanese translation in especial from the machine translation system between Korean and Japanese language which have linguistic properties and similarities strongly in Syntax or Phraseology as the Ural-Altai languages. For its implementation, we have developed syntax based machine translation system KANT designed to be capable of multi-lingual translation.

But, despite the advantageous linguistic facts between both languages as the typical agglutinative languages there are many complicated problems to solve for Korean to Japanese machine translation.

So, in order to survey these problems more sufficiently and to make strong merits of almost same sentence structure more efficient for machine translation, the KANT/ KJ system was designed to expand easily the range of syntactic limitation for word sensing and to improve the functions for semantic analysis.

1. Introduction.

The current state-of-art of the computer technology and the linguistics, especially of computational linguistics leads the machine translation to be one of the most practical and important themes of the present computer application technologies. Its research is now active through the world and several nations pursue the goal of practicalization and commercialization. Improving the quality of machine translation, they are accelerating their efforts to make the machine translation system more economical in common use.

In these respects, June 1983 Systems Engineering Research Center / KAIST(Korea Advanced Institute of Science & Technology) has set up to develop the machine translation system for practical use with the aids of the Ministry of Science & Technology in order to actively meet the need of information-exchanges with other countries. So we have developed the prototype system KANT (KAIST Automatic Natural Language Translation),

and now are applying it to translation between Korean and Japanese.

In this paper, we inform you the result of research/development of Korean-to-Japanese translation, of which practical employment is limited to the scientific and technical fields so far.

In viewpoint of the system skeleton, although we adopted the syntactic transfer method mainly, the translation processor module and the linguistic information module are separable as two independent packages.

The linguistic information module of two languages consists of grammar packets, machine dictionary, and demon procedures. The grammar packet is designed for using Frame Knowledge Representation method, each CASE frame of it can analyze case information and determine word classes or obtain partial semantic informations.

The machine dictionary can contain the lexical, morphological, case informations, the category symbols or semantic features, the equivalent Japanese informations for each word, and the partial grammatical rules if necessary. The Demon procedures are used mainly for inflection of verbs and adjectives, and variation of particles.

At present the system can translate some Korean texts, the current research is focused on the improvement of the translation grammar and the machine dictionary, and the handling of multi-meaning words for expanding the domain of semantic analysis.

2. Linguistic Properties and Design Strategies

2.1 Linguistic properties of Korean and Japanese

(1) Linguistic Comparison of Korean and Japanese

The assertion that Korean and Japanese belong to the same linguistic family has been made by Korean, Japanese, and European scholars from early times. That Korean is an Ural-Altai language is supposed and proved and almost established by comparative linguistic method. In the case of Japanese, however, theory of Ural-Altai language and theory of south family language are competing now.

But whichever class Japanese belongs to, we can find that Korean and Japanese have common properties of Ural-Altaics language, especially of Altaics language as follows :

- . there is no concept of sex in grammar.
- . postpositions are used instead of prepositions.
- . conjugation of verb is well developed.

Besides these common properties, employment of Chinese character and word order are similar in both languages. In addition the phrase is, in both languages, constructed with lexical morpheme followed by grammatical morphemes. Moreover syntactic similarities between both languages are as follows :

- . the word order is free except for a predicate (verb or verb phrase) located at the end of a sentence.
- . the case of nouns is described by adding the suffix
- . the mood can be made by adding the suffix.

These syntactic and semantic similarities between both languages are helpful to implement Korean-Japanese machine translation. But, in practice, there are many problems in implementing machine translation such as of homonymity, ambiguity, inflection and syntactic structure in a specific phraseology.

(2) Korean alphabet HANGUL and Japanese alphabet KANA

Korean alphabet HANGUL consists of phonetic symbols which each symbol is a phonemic transcription. There are about 50 phonemic transcriptions in Korean and the method of composing Korean syllable is shown in Fig 2.1.

A syllable is formed by arrangement of some phonemic transcriptions and a word by a syllable or some syllables.

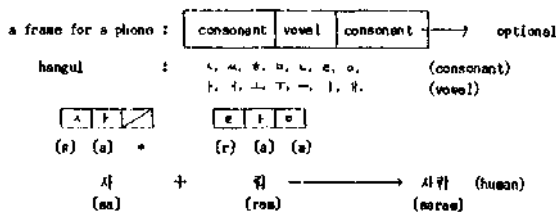
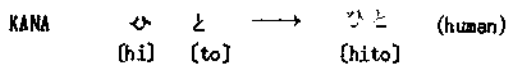


Figure 2.1 Phonetic transcription in Korean

In other hand, KANA can be also called phonetic alphabet but each symbol in KANA describes a phonetic transcription. There are about 50 phonetic transcriptions in Japanese and a word is formed by arrangement of some phonetic transcriptions.



2.2 Application to Korean-Japanese Translation

The similarities in the sentential structure between both languages, Korean and Japanese, and the

identity in the agglutination sequence constructing a sentence with its constituents must be the quite important findings to be significantly considered in the process of their machine translation .

In general, as many linguists interested in these languages accept the opinion that both Korean and Japanese belong to the Altaic language family having some identical characteristics of as the typical agglutinative language, the machine translation between both languages has quite a lot of advantages over other languages in several basic aspects such as especially in the agglutinative patterns of words, [S + O + V] sentential structure in syntax, and use of Chinese characters in their sentence .

Accordingly , the goal of syntactic /case analysis for machine translation From Korean to Japanese (or Vice versa) must lay stress on taking out right one from the multifunctions of grammatical component in every phrase or clause, and on fixing exact meaning of words rather than on transforming syntactic structure of source language into that of target language. That leads us to the fact that analyzing the grammatical multi-functions of some morphemes and words, and extracting the exact meaning from the ambiguous words or phrases without harming the original meanings of sentence are more important than any other things in Korean to Japanese machine translation . As a result, we should effectively make out language dependent knowledge from analyzing linguistic phenomena of both languages and adopt the grammar writing system which is capable of performing syntactic transfer and application to pattern matching covering each linguistic difference. The Korean - Japanese translation system designed under this kind of methodology urges us to pay more attention to the construction of machine dictionaries. We have constructed every execution module as the Figure 2.2. , and processing subject in each module is for morphological and syntactic/ case analysis of Korean, syntactic transfer from Korean to Japanese, syntactic/case synthesis to Japanese, and morphological synthesis for producing the target sentences.

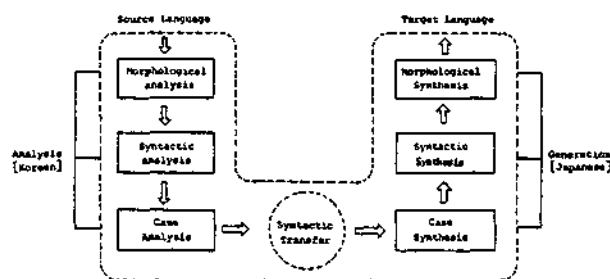


Figure 2.2 Diagram of translation process

(1) Morphological Analysis

Every phrase, Composing a Korean Sentence as source language, is divided into lexical component and grammatical one so that it can be useful for syntactic/case analysis. Lexical component which can be categorized largely into nouns or stem of an inflected word can give us its word category information and grammatical attributes in it by referring to the morpheme dictionary. Grammatical component which can mostly consist of a particle (post-positioned) and the endings of inflection word gives us every information of inflected pattern, case information, interrelation with lexical components, and syntactic feature.

(2) Syntactic / case Analysis

In this step, the system can perform syntactic analysis for multi-functional morphemes and various syntactic features resulted from morphological analysis by using every information that is drawn out of lexical and grammatical components. And then the system can establish case relation with other words / constituents (Prepositioned Nouns or Postpositioned Verbs and adjectives) by analyzing the case information representing stem of the syntactic structure in both languages.

To extract the appropriate target words at the time of syntactic transfer, this module should analyze syntactic or partially semantic features of each word and syntactic interrelations with other grammatical components and words in Korean sentence.

Making this process in right application requests us to use translation grammar capable of linguistic knowledge representation having Frame structure and to build up and analyze the morpheme /Syntax informations such as inflection type, case pattern, syntactic and semantic features.

(3) Syntactic Transfer

Korean texts resulted from syntactic / case analysis have the intermediate sentential form - node list with interlingual labels - for synthesis to target texts in Japanese. Here, for machine translation between both languages having the same syntactic structure, the system should clear the syntactic and semantic relations more in sentential structure, based upon interrelations between [NOUN], [Case(particle)], and [Inflected word] to generate semantic informations for target texts. So the linguistic and partial word knowledge necessary for syntactic transfer to Japanese sentence must be described in grammar rules with the Frame structure.

(4) Syntactic/case Synthesis

Through this step, the system can synthesize the translated informations selected from the

former step into words, phrases, clauses, and sentences.

(5) Morphological Synthesis

The translated words and morphemes equivalent to the each component of source text are synthesized into target sentence corresponding to the structure of source sentence. And the lexical and grammatical components are combined to fix inflected endings of target Verbs (or Adjectives) by referring to the inflection pattern of Verbs in source sentence. Especially, morphological informations or like that (for example, inflected word-type, varied particles) having ties with transformation between both languages can be extracted by using the Demon procedures.

This system - for Korean to Japanese machine translation - demand that we should classify word classes or morphemes adequately, Put suitable and exact meaning with flexible expressions into the dictionary. And in order to translate some parts of Korean sentence concerned with Voice/Aspect/ Tense efficiently, we should scrutinize the agglutination properties of constituents of sentence, classify minutely syntactic categories or semantic features of words especially in nouns, verbs and adjectives. Besides these problems, if we research the linguistic phenomena of both languages more closely we can find out that there are a lot of differences in them such as different phraseology, ambiguous word, much more inflection type in Korean rather than Japanese, different usage of case marker and so on. But, despite these complicated problems we will apply the KANT system to machine translation between Korean and Japanese with overcoming each problems step by step.

2.3 System Design Strategies

(1) Portability to General Computers

When we design a simple prototype machine translation system, the coverage and the quality of translation are the main concern rather than the speed and storage requirement of translation. But, although the large-scale, LISP or PROLOG machines can be more suitable for the natural language processing systems such as nowadays machine translation system especially, we should meet the national requirements for the domestic computer industries.

And so there are some limits in selecting the hardware environments, programming language and tools, in designing the translation system consisted of translation

processor, grammar rules, dictionaries, and utility program. Accordingly, we have chosen PL/I and C as the programming languages, and set up to develop the PL/I version system of large scale computer and the C version system of small scale computer available for Korean and Japanese languages processing.

(2) Natural and Intuitive Rewriting Rule

Linguistic knowledges for machine translation are always imperfect. So it should be updated and complemented continuously. The dictionaries and some parts of the grammar should be updated if the application field is changed. Hence the grammar writing language should be designed easily enough for computer-naive users to grasp the overall flow of the grammar at a glance. Simple pattern of a sentence is easier to understand than a complicated parse tree or ATN.

(3) Modular and Flexible Grammar Organization

We have grouped a set of rewriting rules into a grammar packet. And the grammar packets are hierarchically networked so that the control between the grammar packets become efficient and recursive translation of an embedded sentence is possible.

Modular updating, insertion or deletion of the grammar rules are done in a grammar packet. The grammar is interpreted and compiled into an optimized object module to be linked with a core resident object module. As a result, the translation load module can be operated speedily.

(4) No-fail Translation

Korean and Japanese belong to the same language family, the Altaic as the typical agglutinative languages, and have advantages that the word to word translation between both languages in a sense is roughly possible. The sentential part where syntactic transformation is necessary is not global but local.

So Korean-to-Japanese translation is possible for any sentence without harming the meaning of a sentence too much. When syntactic analysis is failed in a bilingual translation which is based on transfer approach, the later procedure of translation in the system will be at a loss, and the translated result will be odd. But the sys-

tem can undertake recursive processing. And we have used semantic markers and case frames to resolve 1:N corresponding words using semantic informations of the sentence.

3. Grammar Description

3.1 Internal Representation of a Sentence

The internal representation of a sentence has a strong effect on the framework of a grammar writing language and the efficiency of data processing. Dependency trees, phrase structure trees, conceptual structures, or case frames are used to express the structure of a sentence. KANT uses a simple list of nodes to express a sentence, which is same as a tree structure logically.

Each node of the list has three labels of properties - source word, target word, and a category symbol. A lot of information such as syntactic category, number, semantic marker, case information, and other things are encoded in the category symbol.

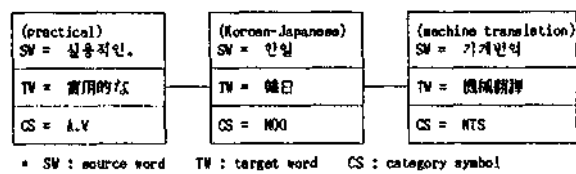


Figure 3.1 An example of the node-list in KANT

3.2 Rewriting Rule

The basic component of a grammar writing language in KANT is a rewriting rule with a following form: (COND(<left>, NEW(<right>, VAL(<op> (<action>, <action2>)...]); (comment)

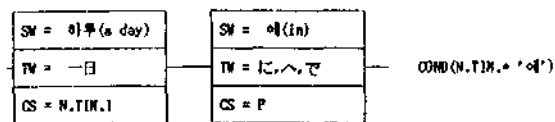
The rewriting rule in KANT transforms one node-list into another node-list until the node-list is transformed into a final sentential form.

We will explain each parameter of the rewriting rule below.

(1) Left

<left> specifies the matching condition of the node-list. It corresponds to left-hand-side of a context-sensitive rule. It allows a grammar writer to specify category symbol patterns or source words to match.

For example, the node-list in which a noun(N) representing a time (TIM) and a particle (?) '에' follows it in Korean is expressed as follows.



Here, the asterisk, a masking metacharacter, will match any character. If a particle such as '은', '는', '이', '가' follows a noun phrase in Korean, they form together a subjective phrase. It can be expressed in one rewriting rule as follows.

COND(N.***.*('은', '는', '이', '가'))

Rewriting rules can be used with pure program lines. If a grammar writer wants to write an exceptional matching condition that is hard to describe in <left>, he can express it directly in program lines before the rewriting rule as follows.

```
if (transitive == YES) /* transitive verb pred. */
(COND(N.FOO.* '을').....) ; translate to 'GA'
```

(2) Right

The structure and the label values of the transformed node-list is written in <right>. It corresponds to right-hand-side of a context-sensitive rule. The transformed node-list is described by category symbols of the nodes or two digit sequence numbers of the nodes which are used in <left>. A rewriting rule which transforms Korean particle '가' following a noun group (NG) into Japanese particle 'が' is described as follows.

```
Noun group '가' -> Noun group
(COND(NG '가'), NEW(01 GA), ...); GA is Japanese
particle (が)
```

(3) Operation

<op> specifies operations for the node-list which has matched with <left>. It allows a grammar writer to set values to the property labels of the nodes. The label values can be made from the label values of the nodes matched with <left> or can be newly set. The nodes can be inserted, deleted, or updated. <op> specifies the target words setting operations of the nodes in <right> in sequence.

```
(COND(NG '가'), NEW(01 GA), VAL(01 '가'), ...);
      ↑
      |
      | set new value '가'
      |
      |
      | set value from node NG
```

(4) Action

(action)s are optional, and may be procedure names. These are activated when the matching condition of <left> is satisfied and node-list transformation is done. A grammar writer can use it for treatment of ambiguities, table handling, or control of the grammatical rule application.

(5) Comment

In this part, the comment about the rewriting rule is described to the end of the line. This part is omitted later by the grammar rule translator.

4. Control of the Grammatical Rule Applications

The analytical grammar of a natural language is very huge and complex. The grammar of a natural language is another type of a large software and we can also use the concept of division and conquest, top-down design modular design and others in it. The grammatical rules in common property should be grouped to be handled efficiently. KANT allows a grammar writer to divide a whole grammar into several groups of grammatical rules, which we call grammar packets. In grammars with general rewrite

rules (unlike context-free grammars), it is possible for the order in which the rules are applied to have a significant effect on the outcome of a processing. KANT imposes a strict ordering on the entire grammar set by networking the grammar packets.

4.1 Grammar Packet

A grammar packet consists of a packet information part, a set of rewriting rules, and some programs inserted in it.

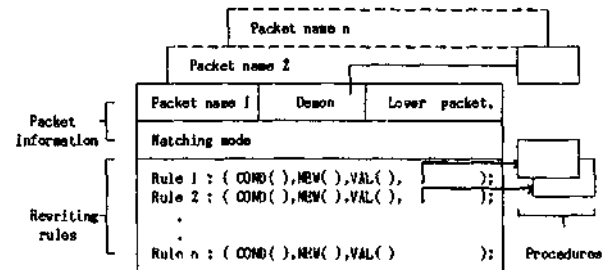


Figure 4.1 The structure of a grammar packet

The grammar writer can specify the matching mode of rewriting rules in a grammar packet with the internal node-list. When a grammar writer specifies LR mode, the matching of rewriting rules will be done from left to right of the node-list one by one. When a grammar writer specifies RL mode, the matching will be done from right to left. When no matching mode is specified, LR mode is assumed.

Rewriting rules in a grammar packet have a priority ordering in their application according to their location. The former one is tried before the later one. The matching sequence of rewriting rules in a grammar packet is as Figure 4.2.

If there is no matching rule in a grammar packet, control flows to the lower packet and the matching of rewriting rules goes on.

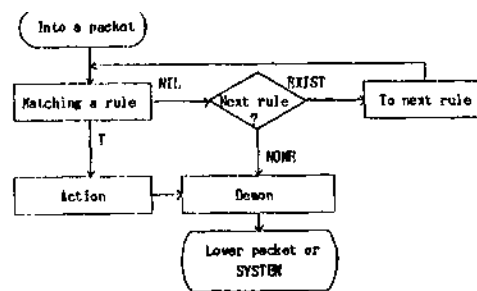


Figure 4.2 The matching sequence of rewriting rules in a grammar

4.2 Grammar Packet Network

A grammar packet network describes the application sequence of grammar packets. Figure 4.3 shows an organization of a grammar packet network.

There is a special grammar packet named SYSTEM in a network. The operation of grammar packets always begins from SYSTEM and goes to the lower packet specified by NEXT-LINK register. So to speak,

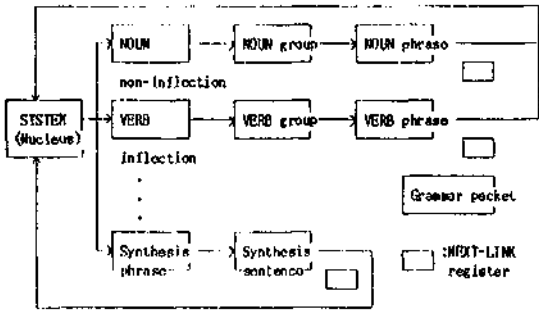


Figure 4.3 An organization of a grammar packet network

SYSTEM is a kernel of control.

A link of grammar packets with SYSTEM packet their head packet is called a packet-link. The last grammar packet of a packet-link does not specify its lower packet, instead it sets NEXT-LINK register to the name of the packet-link to apply next. When lower packet is not specified in a grammar packet, control goes to SYSTEM packet automatically.

After all, all grammar can be seen as a network where a set of packet-links are linked to SYSTEM hierarchically according to their priority of application.

System has a internal match pointer (MP) which points to a node of the node-list. For a given MP value, rewriting rules of a packet-link are tested whether matching with node-list succeed or not. When control reaches the end of a packet-link, control returns to SYSTEM and matching of the rewriting rules of the packet-link repeats with MP points to the next node of the node-list. When a matching of a rewriting rule succeed, node-list transformation is done, actions are processed, and MP returns to the head node of node-list automatically. Control goes to SYSTEM, and application of the packet-link repeats until the node-list transforms to final s-tate.

5. Operation of the Korean-Japanese Machine Translation.

5.1 Operational environment

The Korean-Japanese Machine Translation System KANT/KJ is operated under the unified characters control system which can process Korean HANGUL, Japanese KANA, and Chinese characters. The processing procedure of machine translation mainly consists of the input of Korean-source text, machine translation, and the output of translated Japanese text. Operational environment of Korean-Japanese Machine translation system is as the following Fig 5.1, and now the system has been installed in the national products Super Micro Computer System under the UNIX operating system.

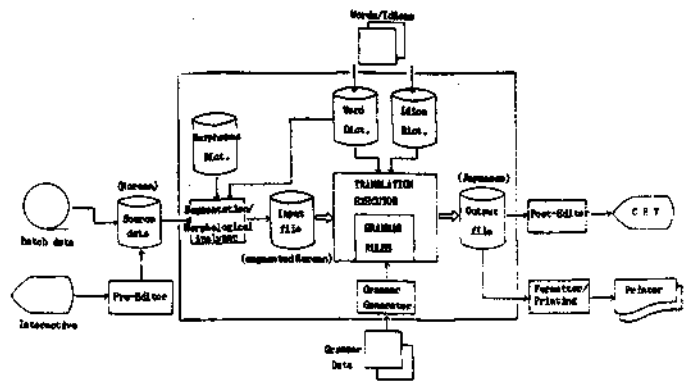


Figure 5.1 Operational Environment of KANT/KJ System

(1) Operational modules for KANT/KJ

a. Input of the Korean text

The source text composed of HANGUL and Chinese characters for Korean-Japanese translation is prepared through the keyboard input of Korean word processing system. Although the commercial Korean word processor in general has every convenient editing function contained with HANGUL to Chinese characters transformation, We have not yet adopted this transformation function to the translation system because of research task to find out how to eliminate the ambiguity of non-inflected homonyms. But Korean editing contained in the word processor will have finally more functions, to transform some parts of HANGUL words to Chinese characters automatically.

b. Machine Translation

The main procedure of Machine Translation consists of the following two modules;

The Automatic Segmentation module: This means morphological analysis of source text. In this module the morphological analysis program separates a sentence into the units of words or morphemes, by referring to some lexicons for analysis of source sentence. And then information of each word or morpheme for the syntactic or case analysis can be reproduced such as the inflected patterns, word classes, grammatical categories etc.

The translation executing module: this module transfers and generates the Japanese text by executing the grammar rules represented with some Frame Knowledges for distinguishing ambiguous meanings. And the module also consists of the translation processor program to control and execute machine trans-

ation, the grammar generator program to compile grammar packets into load module for pattern matching and the linguistic information packages which are composed of several kinds of machine dictionaries.

c. Output of the translated Japanese text.

The translated sentences in Japanese language are produced through the terminal without any post editings, but modified by an editor for processing the KANA especially if necessary. The present results from this system need to be edited in some parts of the texts.

d. system support utilities

There are some utility programs for enlarging the user environments in I/O operation such as dictionary management, PRE/POST editor, File formatter

(2) Translation processing

The translation module of KANT is basically operated with separating from the linguistic information packages. So every linguistic information for making up dictionaries can be easily updated or modified.

The translation module performs the syntactic transfer to Japanese language, by the morphological analysis of segmented Korean sentence, the syntactic and partial semantic analysis with using the Grammar packets represented with frame knowledges for the case analysis. As performing the syntactic transfer, multi-sense and multi-category of Korean words are restricted syntactically, and its corresponding Japanese sentence is produced through the case, syntactic, and morphological synthesis. The practical steps for translation in a simple Korean sentence are as follows:

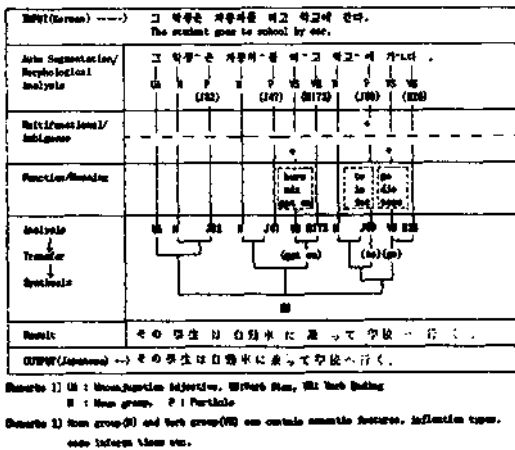


Figure 5.2 Practical Steps of Korean-Japanese Translation

(3) Use of Frame Knowledge Representation

A frame (that is grammar packet in this system) is one of the knowledge representation methods such as predicate logic, semantic networks, scripts, and Mops etc. The grammar packet Korean-Japanese translation is a hierarchical frame network with several levels.

A frame has nodes-'slots' that must be filled with specific data. If its assignment condition is met, a procedure assigns data in it. A frame combines declarative knowledge and procedural knowledge; it is able to handle new situations dynamically. In Korean-Japanese machine translation, a production system is combined with a procedural representation and is systematized into a state transition network. Some semantic rules for analyzing the ambiguous meanings are added in the grammar packets of the KANT system.

5.2 Morphological Analysis module of Korean

(1) Morphological Analysis of Korean

a. Patterns of Korean Phrases

The features of morphological analysis of Korean sentence can be simplified by dividing a phrase into morphological elements because a Korean sentence is composed of phrases which have the formation of lexical components followed by grammatical components.

Fig5.3 shows possible morphological construction patterns Korean phrases.

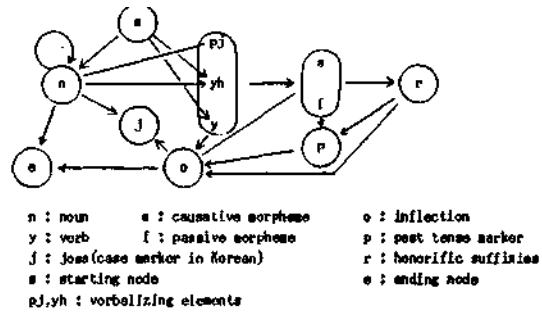


Figure 5.3 Patterns of Korean phrase

b. Inflection of verbs in Korean

When a inflective words-verbs or adjectives in Korean - combine with the inflected endings of a word, there are 11 phonological phenomena in inflection of Korean word. Table 5.1 shows some examples of these phenomena.

class of inflection	condition	example
variation of verb stem	λ inflection λ is deleted before the omi which is starting with 'o'	잇 + 어서 ↓ 이어서
variation of verb stem and omi	ε inflection - + 2 variation of verb stem and 'o' deletion from omi before the omi which is starting with 'o'	흐트 + 어 ↓ 물리

(Table 5.1). Inflections in Korean

d. Phase 4: Translation Success

Japanese text is constructed from the internal node-list. After output of this Japanese text, control goes to phase 1 to process the next sentence.

e. Phase 5: Translation Failure

Error message is displayed and the intermediate result of translation is output to the output file. Control goes to phase 1 to process the next sentence.

f. Phase 6: End of Translation

Terminate the translation procedure.

As the above noticed, the algorithm gives only main frame of translation system and analysis, transfer, or synthesis of a sentence is up to the grammar writer.

(2) Grammar Design

The transfer approach to machine translation allows the system to separate the various translation problems, for example, the source language analysis from the target language synthesis and the parsing strategy from the morphological analysis. KANT Korean-Japanese machine translation grammar is constructed according to the transfer approach (Figure 5.7).

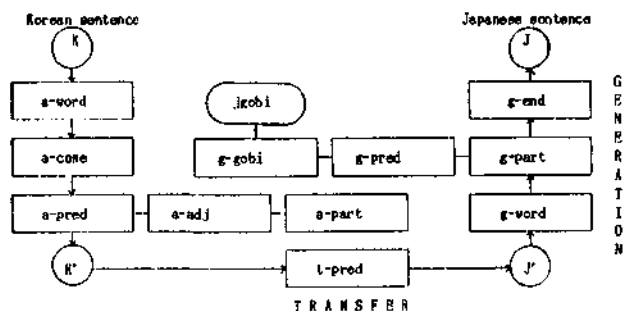


Figure 5.7 Control flow of KANT Korean-Japanese machine translation grammar packets

The "analysis" grammar packets embodies the word-class analysis, case analysis, and semantic analysis of Korean sentences. For example, a Korean word "성숙한 (songsukan)" is analyzed as follows:

성숙한 -----> Korean root : 성숙하(grow)
 Part of speech : Verb
 Inflection : Attributive
 Tense : Past

The "transfer" grammar packet transfer Korean sentence representation to Japanese sentence representation. For example, analyzed representation of Korean word "성숙한" is transferred to Japanese representation as follows :

Japanese root : せいじゅく
 Part of speech : Verb
 Inflection : rentaikei
 Tense : Past

The transfer phase is kept as small as possible. In KANT system, anyhow, we owe many things to the similarity of sentence structures between both languages. In the generation phase, the readable Japanese text is generated from the Japanese sentence representat-

ation produced by transfer phase.

(3) Dictionary

The quality of a machine translation system heavily depends on the quality of the dictionary. KANT Korean-Japanese machine translation dictionary contains the key Korean words, category symbols denoting the properties of the word, the Japanese equivalent words, and some dictionary rules.

KANT system allows a grammar writer to write word specific grammatical rules in an entry of word dictionary. The dictionary rules written in a word dictionary are retrieved with a entry word and saved to be applied later. The dictionary rules are used for making the ambiguities more clear or for idiomatic expressions.

Figure 5.8 shows an example of a part of word dictionary. In this case a-word is the name of a grammar packet where this rule should be applied.

```

문법 0      MPH_      けんなん
위하 0      TM_       ため
위하 1 * (ja(a_word), cond(TM_ '어서는 '_, new(J_), val(a1 '④ ')):
위하 2 * (ja(a_word), cond(TM_ '~ '_, new(J_), val(a1 '④ ')):
(state(a_pred), next(a_adj)): Predicative phrase anal.
(cond(V_ _E004_), new(KG5_ #13_), val(a1)): 2
(cond(V_ _E013_), new(KM5_ #13_), val(a1)): 기
(cond(V_ _E014_), new(KM5_ #13_), val(a1)): 기
(cond(V_ _E109_), new(KB5_ #13_), val(a1)): 어서
(cond(V_ _E111_), new(K45_ #13_), val(a1)): 2
(cond(V_ _E130_), new(KK9_ #13_), val(a1)): 2
(cond(V_ _E145_), new(K05_ #13_), val(a1)): 2
(cond(V_ _E166_), new(KK3_ #13_), val(a1)): 2
(cond(V_ _E167_), new(KK5_ #13_), val(a1)): 2
(cond(V_ _E172_), new(KG5_ #13_), val(a1)): 2
(state(a_adj), next(a_part)):
(cond(A_ _E002_), new(KB5_ #13_), val(a1)): 기
(cond(A_ _E111_), new(KG5_ #13_), val(a1)): 2
(cond(A_ _E116_), new(KK5_ #13_), val(a1)): 2
(cond(A_ _E145_), new(K05_ #13_), val(a1)): 2
(cond(A_ _E166_), new(KK5_ #13_), val(a1)): 2
    
```

Figure 5.8 An example of a dictionary

(4) Translation Result

A sample raw (not preedited and not postedited) translation is like Figure 5.9. On the whole the translation is somewhat rough but readable. Postediting will remove the roughness so that the final printed version is at least comparable to that which would be rendered by a competent human translator.

```

1      *** KOREAN TO JAPANESE MACHINE TRANSLATION ***
2                                     STATE => T
3      ----- INPUT TEXT(KOREAN) ----->< OUTPUT TEXT(JAPANESE) -----
4      성숙한 인격에 이르는 길은 험고도! せいじゅくしたじんかくにいたるみちは
5      険難한 길이다. 사람이 사람다워져! とおくけんなんなみちである.ひとがひ
6      기 위해서는 끈기 있는 노력이 있어야! となしくなるためにはこんきのあるどり
7      아 한다. 그것은 하루 마찰에 성숙! にくがなければならぬ. せいじゅくに
8      되는 것도 아니요. 인격 도마를 위한! ちあきかにせいじゅくされるものでもなく
9       결단과 노력 없이 성취할 수 있는 것! じんかくのとうやのためのけつだんとど
10     도 아니다. 그러기에, 많은 사람이! にくくなしにじょうじゅくされることので
11     성숙한 인격에 이르지 못한 채 그들! みるものでもない. せいじゅくにおおくの
12     의 인정을 따지는 것을 볼 수 있다. | ひとがせいじゅくしたじんかくにいたら
13     성숙한 인격을 이루기 위해서는! にくいままかれらのしょうをおえることを
14     운을 작히고 지식을 추구해야 한다. | みることができる. せいじゅくしたじん
15     인격다운 삶은 세계 많은 지식과 기! かくをなしとげるためにはかくもんをお
16     능을 필요로 하며, 어떤 직업에 있어! ばえちしきをついせうしななければなら
17     서나 그 직무를 수행하기 위한 능력! ない. じんげんらしいしょうはたいてい
18     을 필요로 한다. 그러므로 성숙한! ぶおくのちしきと善のうをひつようとし
19     인격을 꾸미기 위해서는 우선 무지! どのようなしよくぎょうにおいてもその
20     에서 합리하는 일이! 절실하게 요구! しくむをすいこうするためののうりよ
21     원다. | にくをひつようとする. せいじゅくしたじんかくをぐびするたためにはま
22     | ずむからだっぴすることがせつじつに
23     | ようきゅうされる.
24
    
```

Figure 5.9 A sample raw Korean-Japanese translation

6. Conclusion

A machine translation system KANT and its application to Korean-Japanese translation have been introduced in this paper. The characteristics of this system can be summarized as the following items :

(1) Major processing of the system is based upon the syntactic transfer, and some grammar packets represented with the frame knowledges for resolving the ambiguous meanings have been assigned to the system.

(2) Machine dictionaries and translation module are separable, and both modules are linked as one operational system under translation.

(3) Every procedure for translation consisted of the Korean analysis, Korean to Japanese structural transfer, and Japanese generation is integrated in the same packet.

(4) Sentential expression is node-list, and a rewriting rule is an expression in the form of node-list transformation. It is possible to insert some programs and write the rewriting rules in a grammar packet for powerful and flexible expressions.

But, despite many similarities between both languages in the syntax and expressions we sometimes need post-editing some parts of the output sentences in order to get the more pliable, exact result from the system.

Though someone has said it seems simple to accomplish a development of machine translation between Korean and Japanese because of the agglutinative language with the most similar syntax, we should recognize the facts that there are lots of problems to solve especially in the points of the ambiguous meanings of some words, contextual meanings of the texts, the different expression or nuance from the society using other lingual pattern, and some differences or varieties of the lingual usages. So we have adopted the grammar packets represented with frame knowledges for extracting exact meaning from the ambiguous words and generating the pliable Japanese sentences more efficiently. But we sometimes are not satisfied to have bad results from the system, especially in multivocal words and contextual sentences. And so the research for solving the above difficult problems should be continuously undertaken. Although its applicable range at present is limited to the Scientific & Technical area, the capability of the present system is in the level of selecting the suitable meaning from ambiguous word especially in multifunctional particles, some nouns and verbs.

We are currently improving every factor of the system and upgrading the linguistic modules such as grammar packets with frame knowledges, dictionaries and its subsidiary rules, and demons for realizing practical use within the specific domain.

[REFERENCES]

- Sawai, S., et al., "Knowledge Representation and Machine Translation", Proc. COLING82, 1982.
- Nagao, M., et al., "Grammar Writing System (GRADE) of Mu/Machine Translation Project and its Characteristics", Proc. COLING84, 1984.
- Boitet, Ch., et al., "Implementation and Conversational Environment of ARIANE 78.4", Proc. COLING 82, 1982.
- K.S.Lee, et al., "A study on the development of machine translation system between Korean and Japanese." Korean Ministry of Sci. & Tech., 1984-1985.
- C.H.PARK, et al., "Automatic leaving Spaces and meaning analysis of the Case by using the FKR in Japanese-Korean Machine Translation ." in Proc. of KIEE, Seoul, 1984
- J.T.Lee, et al., "A Syntax-based Multilingual Machine Translation System", J.Korea Inform. Sci., Vol.13, No.1, 1986.
- Bruce, Bertram, "Case systems for natural language.", Artificial Intelligence 6:4, 1975.
- J. Carbonell, et al., "A Tutorial on Techniques and Applications for Natural Language Processing." Carnegie-Mellon Univ., 1983.
- Hayes, P.J., "The Logic of frames, Frame Conceptions and text understanding." Walter de Gruyter and Co., 1979
- Colby, K.M and R.C. Parkinson, "Pattern-Matching Rules for the recognition of Natural Language Dialogue Expressions." American Journal of Computational Linguistics I, 1974
- Cercone, Nicholas, "Morphological Analysis and Lexicon Design for Natural Language processing." computers and the Humanities 11, 1977