[*International Conference on the State of Machine Translation in America, Asia and Europe*. Proceedings of IAI-MT86, 20-22 August 1986, Bürgerhaus, Dudweiler]

English-Japanese Interactive Translation System: LUTE-AID

Hirosato NOMURA
NTT Basic Research Laboratories
Musashino-shi, Tokyo, 180, Japan

Hitoshi IIDA
ATR International
Twin 21, Higashi-ku, Osaka, 540, Japan

1. Introduction

We have been developing a series of experimental machine translation systems called LUTE (Language Understander, Translator & Editor) [1] to support the examination of our theoretical work on computational linguistics. Since we are mainly interested in the theoretical aspect of natural language understanding, our objective is not to develop practical systems but to develop fundamental theories applicable to the design of a high ability machine translation system [2]. There is no question, however, that some of these experimental systems may in fact provide practical application to our daily work as tools for manipulating large amounts of linguistic and non-linguistic data as well as for examining such data through programs. We have thus been focusing part of our effort on developing a computer environment we collectively call the Reciprocal Environment for the Study of the Language Understander, Translator & Editor (RESOLUTE) which consists of many facilities for supporting text analysis, dictionary maintenance, and program and knowledge-base development and testing [3],[4].

This paper presents basic ideas called Pattern-driven Translation (PDT) adapted to the design of LUTE-AID which is an English to Japanese interactive translation system in the LUTE series. In the beginning, LUTE-AID was a small function in RESOLUTE, providing the means to access computerized dictionaries and examine verbatim translations of small sample sentence fragments. Later, it became an independent system, and then it naturally began incorporating the characteristics of an interactive translation system. Therefore, LUTE-AID is not a well-designed translation system intended for practical use, but is simply a kernel to which extra modules to examine specific problems arising in the research can be added easily. After such experiences, LUTE-AID has become an interactive translation system in the LUTE series, however, its design and operating principle is quite different from other systems [5]-[9] in the series.

The LUTE-AID prototype was developed on a DEC2060 in October 1984 and revised in January 1986. Its current program size is 120KB in total. It is operated on semi-graphic terminals that can display KANJI and KANA as well as alphanumeric characters and symbols. Recently, also a version for a VAX-11 has been developed.

This paper begins with a description of overall characteristics of LUTE-AID in section 2. Section 3 presents a scenario of a translation process, which enables the interaction process to be clearly understood. Section 4 describes the grammar utilized by LUTE-AID. The grammar employed for analysis is represented by a representative framework called Syntactic Configuration Pattern (SCP), and that used for transfer is represented by a similar framework called Configuration Concoction Pattern (CCP). Section 5 describes the processing adopted by LUTE-AID. In the processing, the analysis process is referred as Pattern-driven Analysis (PDA), and the transfer process is referred as Configuration Concoction Transfer (CCT). Section 6 describes the lexicon utilized by LUTE-AID, and finally, section 7 describes translation experiments conducted by LUTE-AID on English textbooks for Japanese students.

2. Characteristics of LUTE-AID

LUTE-AID is typical of Machine-Aided Translation system; it performs a rough draft and the user has the responsibility to finish the translation. Verbs, or predicates, if strictly speaking, are translated by allowing the user to select the most suitable word from a list of choices, while other words are automatically translated by the system. This is because verb selection is thought to be the key in determining meaning, and, as such, is a crucial factor in translating between such languages having very different structures as English and Japanese.

LUTE-AID mainly utilizes syntactic information in the processing and its processing ability is suppressed as low as possible to ensure simple system construction. Conversely, the main LUTE systems [4],[6],[7] primarily utilize semantic information and analyze deeper sentence structure meaning by utilizing knowledge-bases [10], accordingly, they have complicated system constructions. However, since LUTE-AID comprises only the minimum fundamental functions needed for translation, and since its program is highly modularized, it is easy to add other modules to extend its translation ability to include, for example, deep semantic processing as mentioned above.

When the system recognizes that a word is not contained in the lexical entries, it asks the user questions so that it can learn certain lexical information including the translational equivalences of the word. At that time, the system also asks whether the user would like to store the new information into the lexicon. This enables adding a new lexical entry or modifying existing lexical items. Thus, the system provides an on-line facility for maintaining the lexicon.

The interaction then continues until sentence translation is complete. LUTE-AID usually produces more than one translation from one source sentence, and the user must then select the best one from among them. After this stage, the system provides an extra facility for changing translations for words other than verbs. Additionally, the system provides a multi-lingual word processing facility, including KANA-KANJI conversion, for post-editing of the translation draft. This is made available when the user cannot finish the translation by changing the translations of words. This facility also provides a means for changing the word order. If every translation candidate is definitely unacceptable, or if direct human translation seems to yield a quicker result than the modification of unproper candidates, the user himself can freely type in the entire or partial translation as he sees fit. Although such a facility appears strange for an "automatic" translation system, it is obviously very useful when examining a translation or acquiring new lexical and grammatical information during research.

The system allows computerized dictionaries to be displayed on the same screen for help when the user feels the lack of linguistic knowledge or skill for post-editing or direct translation. Thus sample translations can be referred or copied into the draft translation. Moreover, it is easy to imagine that a grammar book, for example, can be computerized as a file so that it can be displayed on the same screen simultaneously. Such a system can surely be seen as a translation expert system in terms of so-called artificial intelligence. However, the current system has not yet been equipped with such a computerized grammar book.

LUTE-AID has only one bi-lingual lexicon commonly used during analysis, transfer and generation. Each lexical item has only a little information about syntactic categories and translation equivalents, and a small amount of additional semantic information. Therefore, it is very easy to add new lexical entries and items freely without bothering to maintain consistency. Since the user is allowed to assign the name of the lexicon to be utilized in the translation process when programs are loaded or set up, the user can have a private lexicon. Thus the user can utilize a huge sized lexicon covering almost all words, or a small sized lexicon covering only a special domain and thus containing only the domain-specific translations.

Apart from its size, the LUTE-AID lexicon consists of several levels of entry complexity concerned with the compounding words. The most fundamental level is for isolated words. Other levels concern compound nouns or combinations of idiomatic expressions. Since a large lexicon necessitates considerable memory space and computation time, the system permits the user to assign a desired system utilization lexicon level. Additionally, it has some computerized dictionaries as mentioned above. They are not the lexicon utilized in the translation process, but computerized dictionaries dedicated to human use. They include Japanese, English, Japanese-English, and English-Japanese dictionaries, and all have more than sixty thousand entries.

Interaction is carried out through a display screen that is separated into three regions, one for source text, one for target text, and one for communication between the user and the system. The source text may be either read in from a disk file or directly typed in from a keyboard. The target text is displayed on the display and stored in a disk file. The communication region is multi-purpose for inputting commands, displaying translation candidates, and executing word processing, etc. These three regions are resident on the screen while one more extra temporal region dedicated for displaying computerized dictionaries can be made to appear. Facilities for the interaction look like traditional ones since the current system does not use the high resolution bit map display which is available, for example, on Symbolics Lisp machines. There is no reason why

LUTE-AID has not yet been implemented on the Symbolics, since the task of writing the programs for the present system was just practice for a person who first wrote Lisp programs.

The strategies for the overall translation process are called Pattern-driven Translation (PDT), which consists of Pattern-driven Analysis (PDA), Configuration Concoction Transfer (CCT) and Pattern-driven Generation (PDG). The PDA process utilizes grammatical information and word functions formalized simultaneously as Syntactic Configuration Patterns (SCPs), and the CCT process utilizes syntactic transfer rules and word-oriented concoction procedures formalized as Configuration Concoction Patterns (CCPs). The representative framework for an SCP also provides a representative framework for representing the analysis result, and that for a CCP has almost the same representative framework. Thus, the PDA process produces SCP representations of a source text, and the CCT process then transfers the source SCP representations into target SCP representations, and the PDG process finally produces target texts from the target SCP representations. Syntactic information incorporated into the SCP and the CCP is based on constituent configurations. Its representative form is a pattern so its process is the so-called pattern-driven processing. These patterns can not be modified during the translation process in the current system.

3. Scenario of interaction in translation process

Figure 1 shows the flow of pictures on the screen during the translation process. The upper region is for the source text, the middle region is for the target text, and the lower region is for communication as mentioned above. Figure l (a) shows an example of source text just typed in for translation. The sentence is well known as a burdensome sentence for syntactic analysis since it has many possible syntactic analyses. However, this scenario is not intended to show how to analyze such a sentence but it is only to show how the interaction is carried out.

In the beginning, the system asks which level of the lexicon the user assigns. The questions concern what lexicon level should be utilized. The questions are made in the form of asking whether each lexicon level of phrasal verbs, compound words, phrasal adverbs, phrasal adjectives, and phrasal prepositions is to be utilized. Figure l (b) shows that the user assigned the lowest level of lexicon utilization since all answers were 'no'. Therefore, the system was now limited to using the most fundamental lexicon level.

After setting up the lexicon utilization level, the system starts the translation process. The communication region in Figure l (c) is asking which translation, from among four different translation candidates, the user will prefer for a verb 'like'. A part of the translation for the source text has already displayed in the upper region for words other than the verb, which will help the user to select the most appropriate one for the verb. As such, usually, three or four candidates are enough for the translation of domain-specific text. However, some lexical items in the computerized dictionaries have more than fifteen translations with which to amaze the user. Thus, the user is responsible to finish the translation when translating by LUTE-AID. It is true that the selection of the best translation, i.e., one to which all people will agree, is incredibly difficult with the current technology.

As mentioned above, the system always asks for the selection of the translation for predicates. The class of predicates includes not only verbs but inflected verbs and adjectives in some special usage. Here, the translation candidates are as follows: (Hereafter, the italics following the Japanese is the corresponding Romanized expression of the Japanese.)

1) 好む *konumu* (This is a verb and its equivalences are 'like', 'be fond of', 'prefer', etc.),

2) し たい と 思う *shi tai to omou* (This consists of a verb, an auxiliary verb, a particle, and a verb. Its equivalence is, for example, 'want to do'),

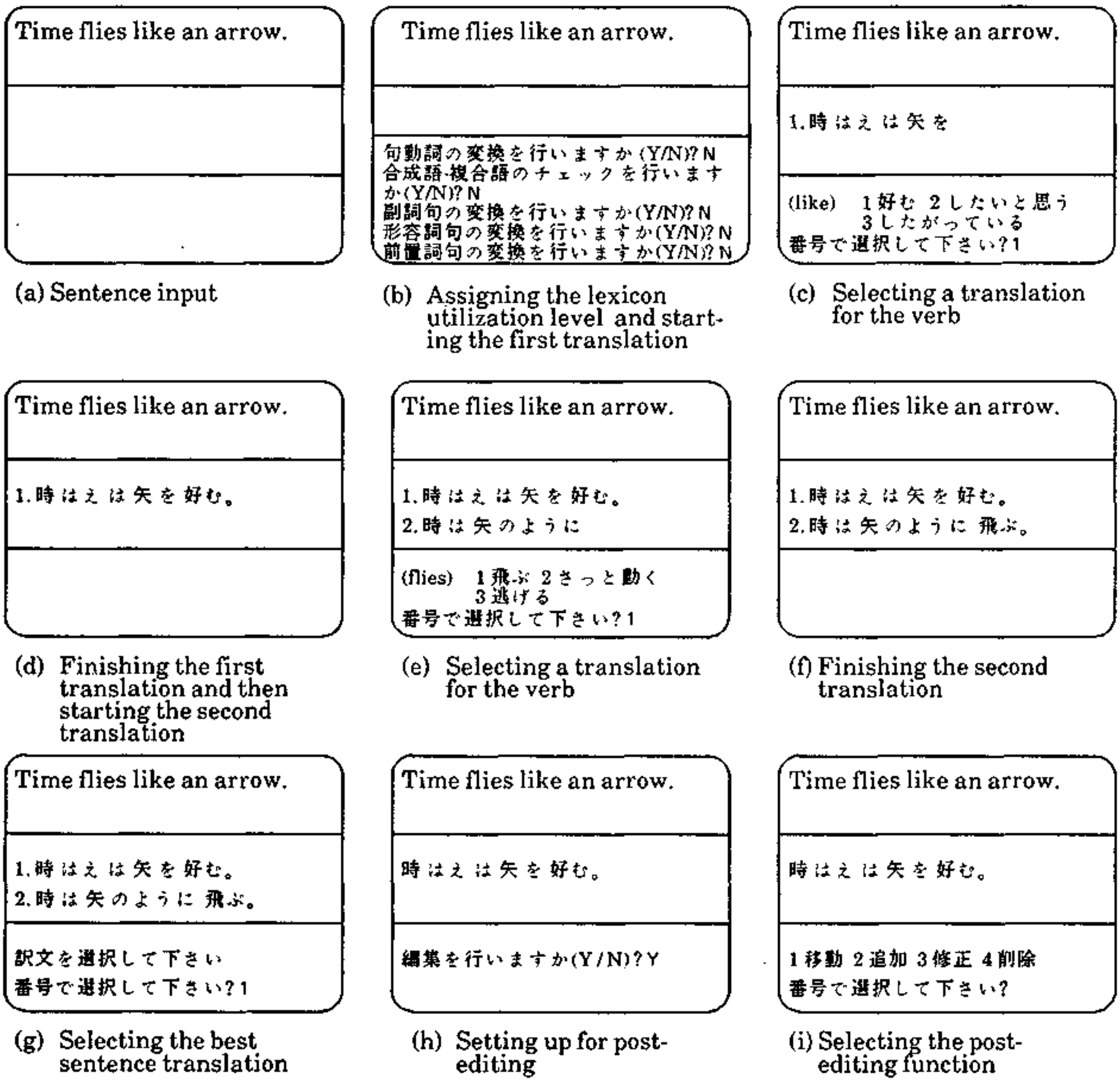

Fig. 1 Screen interactions during translation process

3)したがっているshi taga tte iru (This consists of a verb, an auxiliary verb, a particle, and a verb. Its equivalence is, for example, 'eager to do'.)

By selecting 1, the first translation is produced as shown in Figure l (d):

T1) 時はえは矢を好む*toki hae ha ya wo konomu,*

where, '時はえ*toki hae'* is a compound noun consisting of two nouns '時*toki'* and 'はえ*hae'*. '時 *toki'* is equivalent to 'time' and 'はえ*hae'* is equivalent to a 'fly' (an insect). 'は *ha'* is a particle and expresses a theme in the sentence. '矢*ya'* is a noun and equivalent to 'arrow'. 'を*wo'* is a particle and expresses the object case of the sentence. '好む*komomu'* is a verb. Therefore, the system understood that 'time flies' is a compound noun and is the subject of the source sentence, 'like' is a verb, and 'an arrow' is a noun phrase with an article 'a' and a noun 'arrow', then is the object of the verb 'like'.

Since one target translation has already been produced, the system begins the second try. Figure l (e) shows the selection of the translation for a verb 'fly' inflected for the third person and the singular noun 'time'. Translation candidates are as follows:

4)飛ぶ*tobu* (This is a verb and is equivalent to 'fly', 'soar', 'flatter', 'be flown away', 'be on the wing', etc.),

5)さっと動く*sa tto ugoku* (This consists of an adverb and a verb, and its meaning is, for example, 'move quickly'),

6)逃げる*nigeru* (This is a verb and equivalent to 'run (away, off)', 'flee', 'fly', 'take to flight', 'escape', 'elope', 'break loose', 'evade', 'back out of', 'put off', etc.)

By selecting 1, the second translation becomes as follows:

T2)時は矢のように飛ぶ*toki ha ya no youni tobu,*

where, '時*toki'* is a noun,'は*ha'* is a particle,'矢*ya'* is a noun,'の*no'* is a particle,'ように*youni'* is an auxiliary verb, and '飛ぶ*tobu'* is a verb. So the system understood that 'time' is a noun and is a subject of the sentence. 'flies' is a verb and is inflected for the single third person subject. 'like' is a preposition, and 'an arrow' is a noun phrase.

For the source sentence, the system generated only these two target sentences as shown in Figure l(f) without being bothered by the many other possible syntactic analyses as seen in textbooks concerning context free parsers. Of course, this examination was carried out under the assumption that 'time' is a noun. However, if the 'verb' is added to the part of speech for the word 'time', it is sure that the system will produce more translations.

After finishing the translation trials, the system asks which translation the user prefers as a whole, as displayed in Figure l (g). If selecting 1, for example, now the system asks whether the user wants to post-edit it, as in Figure l (h). Post-editing consists of changing word orders, typing in translations of words or phrases the system cannot produce, changing translations of words other than predicates, and deleting some parts, which are shown in the communication region in Figure l (i). Also referring to the computerized dictionaries is one of the post-editing facilities available, but no picture for that is provided here. To support the typing in of Japanese phrases, a KANA-KANJI conversion facility is provided as mentioned above.

4. Grammar

The grammar utilized in processing is represented by a framework called the "pattern" which represents both the syntactic configuration of a constituent and the lexical functions of the head word of the constituent. As indicated previously, the specific pattern used for analyzing a sentence structure is called the Syntactic Configuration Patterns (SCPs). The head function provides the grammatical role and semantic information for the constituent and is utilized to integrate the constituent and other constituents into one larger constituent. The pattern termed the Configuration Concoction Pattern (CCP) is used for the transfer process. The CCP relates a source SCP representation with a target SCP representation with lexical concoction procedures which provide the information necessary to first rearrange the word order and to then generate a target SCP representation.

The SCPs are provided for classes of constituents. The current number of SCP classes is 20, and they are ADJective PHrase, ADVerb PHrase, AUXiliary PHrase, INFinitive PHrase, Noun Phrase, PREpositional PHrase, RELative ADVerb, Relative PROnoun, Unit SENTence, predicate verb, Complex SENTence, COMpound SENTence, Coordinate PAIR SENTence, VEN Past participle PHrase, VING present Participle PHrase, VING gerund PHrase, EMbedded Unit SENTence, AUXiliary verb TERM (ex. 'ought to'), SUBordinate CONJunction TERM (ex. 'as if), and SPecial TERM (ex. 'in order to'). Each pattern includes some word usages for the head word in the constituent. For example, the ADJPH pattern for an adjective phrase involves the simple use of an adjective like 'beautiful' in a sentence 'the flower is beautiful', or the phrasal use of an adjective-preposition pair such as 'similar to' in a sentence 'it is similar to that.' The verb patterns follow Hornby's verb patterns although they have been extended to distinguish the differences of the more detailed configurations needed to produce better translations. In addition, the verb patterns, and therefore the sentence patterns, have been extended to cover complex sentences, coordinating conjunctions, and relative clauses. Indeed, such fundamental grammatical structures might be sufficient for writing ordinary technical documents if writers and readers do not have *a* need for more elaborate sentence style. This is not saying that all documents should be written in as elementary style, but simply that documents written using clear and straightforward structures will add to cost saving even in computer processing.

The SCP's components consist of some items involving constants, obligatory variables and optional variables, which can be accounted for as follow.

1) constants, where each constant consists of a sequence of letters: abc ... n,

2) single-value variables, where each variable has a single value of a word category or the word itself: >X,

3) multi-value variables, where each variable has a list of values or a word sequence: + X,

4) restriction variables: (RESTRICT +X f1 f2 f3 ... fn), where 'RESTRICT' specifies that the value of the variable +X is restricted by the following conditional functions, f1, f2, f3,..., fn,

5) optionality: if items 1-4 are enclosed with parentheses such as '(a)', '(>X)', '(+ X)', or '((RESTRICT + X f1 f2 f3 ...))', then they are optional in terms of SCP representation.

By adopting these items to describe the SPCs, examples can be as followings.

(i) An SCP of a noun-phrase:

((>DETerminer) (>ADJective) (>Noun-l) >Noun-2).

Here the head noun, >Noun-2, is represented as an obligatory variable having a single value (the value is a word category), while the premodifier describes the sequence of the three optional variables, (>DETerminer) (>ADJective) (>Noun-l), all of which also have their own respective single values (word categories). Since each variable is identified by part of speech, the variable names can be readily utilized to represent each syntactic role, which is convenient for the user to develop and maintain SCPs.

(ii) An SCP of a unit sentence having an intransitive verb and an infinitive phrase:

```
((RESTRICT  +NP-1 NP-test)
 (RESTRICT +I3 I3-test)
 TO
 (RESTRICT   +INFinitive-PHrase  INFPH-test)
 >ENDMARK).
```

Here, 'NP-test' is a conditional function for testing and constructing an NP structure, 'I3' is one of the predefined categories for a set of intransitive verbs, each of which is followed by a 'to-infinitive', and 'I3-test' is a conditional function for testing and constructing a verb structure. The current number of classes for verbs is 7, and each class has several subclasses. The total number of subclasses is 25. Therefore, SCPs for unit sentences are divided into 25 classes according to the verb subclasses. The current system provides 99 conditional functions in total. 'TO' is a variable whose value is bound to the preposition 'to'. 'INFPH-test' is a conditional function for testing and constructing a 'to-infinitive phrase', and 'ENDMARK' is a variable whose value is bound to the period mark. An example sentence for this type of verb use is 'He failed to reach the shore.'

(iii) An SCP for a complex sentence:

((RESTRICT + SUBORDinate-conjunction SUBORD-test)
(RESTRICT +Unit-SENTence-l USENT-test)
> COMMA
(RESTRICT +Unit-SENTence-2 USENT-test)
> ENDMARK),

where 'SUBORD' is bound to a subordinate conjunction, 'USENT' is bound to a simple sentence, and 'COMMA' is bound to a comma.

LUTE-AID offers 78 syntactic categories for describing the SCPs. Categories other than parts of speech can be described by other SCPs with part of them being given as:

<syntactic categories for sentences and clauses>

CSENT ;a complex sentence or a compound sentence,

USENT ;a simple sentence or a main clause in a CSENT,

EMUSENT ;an embedded sentence, or a coordinate or subordinate clause in a CSENT,

CPAIRSENT ;a sentence with no subject in an EMUSENT.

<syntactic categories for phrases>

ADJPH ;an adjective phrase [ex. very beautiful / taller than I],

ADVPH ;an adverb phrase [ex. out / here / in this room],

AUXPH ;an auxiliary verb [ex. I can swim. / You should have completed it before summer.],

INFPH ;an infinitive phrase [ex. I came here to see her.],

NP ;a noun phrase [ex. a bread knife / a comfortable house to live in],

PREPH ;a prepositional phrase [ex. at the door / to the boy],
VENPP ;a past participle phrase [ex. He got trapped.],
VINGPH ;a gerund [ex. I enjoyed singing.],
VINGPPH ;a present participle phrase [ex. He came running.]

Although conditional functions mentioned above ('NP-test', 'PREPH-test', etc.) presented syntactic information, they are also capable of incorporating semantic information. One example is as follows. A conditional function can represent the restriction in the co-occurrence relationship between verb I3 and the dominated cases. Such semantic information is described by semantic features. Moreover, the restricted variables can provide semantically conditioned functions like a variable having a conditional function describing a selectional restriction such as a NP should be animate. Since any number of tests can be added to the restricted variables as functions, an SCP can become a pattern dedicated to a simple sentence whose subject must be animate if a function restricting to have a feature of animate is added. An example of such a restricted variable is

(RESTRICT +NP-l NP-test ANIMATE-test),

where 'ANIMATE-test' is the function added. The number of features utilized for specifying semantic functions is 32 in the current system.

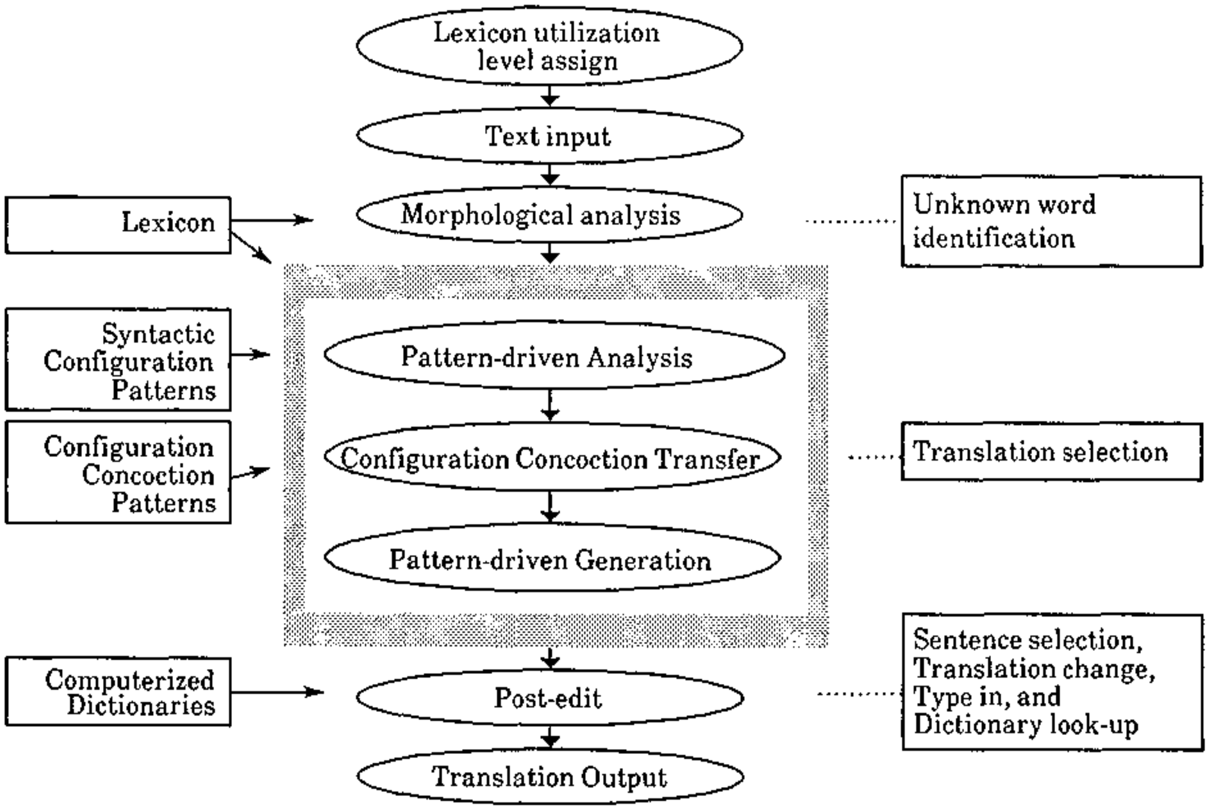

Fig. 3 Process flow in LUTE-AID

5. Processing

We call the analysis process Pattern-driven Analysis (PDA) since each analysis of sub-processes is activated by the pattern directly matching a fragment of a sentence to a SCP. Since the analysis results are also represented by the same framework for an SCP, each analysis result can be seen as an instance of the SCP. Such analysis results are called SCP representations of the sentence. Thus, the transfer process converts the source SCP representation into a target SCP representation, from which the generation process can produce a target sentence. Since the transfer process can utilize word-oriented concoctions procedures, it is called Configuration Concoction Transfer (CCT).

The pattern matching process itself is very simple, thus realized by a small program. However, if the process is programmed in naive manner, it takes considerable computation time. Since both the program size and the computation time are crucial for practical use, thus some heuristics are attached to head functions and concoction procedures. However, there is a trade off between the program size and the computation time. The entire process is presented in Figure 2.

(1) Phrase analysis

For a noun phrase, 'a red arrow', an SCP whose configuration is 'DET ADJ NOUN' is adopted successfully. The PDA process is used to simply match the configuration to the sequence of parts of speech in the noun phrase, then produces an instance of the SCP as an SCP representation by binding all variables with words appeared in the noun phrase. In the process, the restrictions are tested simultaneously. These analysis principles are almost the same as those for analyzing other constituents. The tree representation of the SCP representation of the example noun phrase is shown in Figure 3.

(2) Simple sentence analysis

The analysis process for the simple sentence including the intransitive verb I3 as exemplified formerly by the sentence 'He failed to reach the shore' utilizes the SCP

((RESTRICT +USENT-1 USENT-test)).

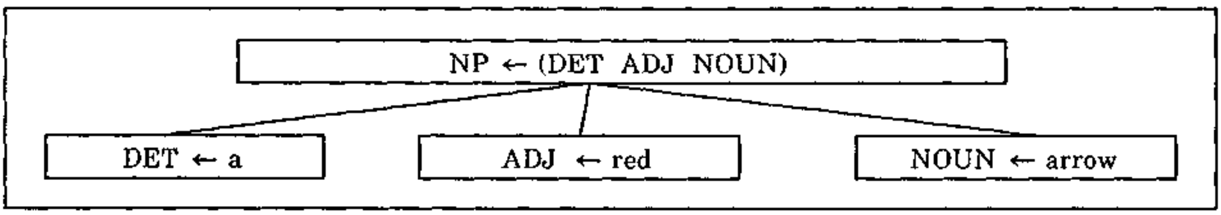

Fig. 3 Example of a noun phrase tree structure

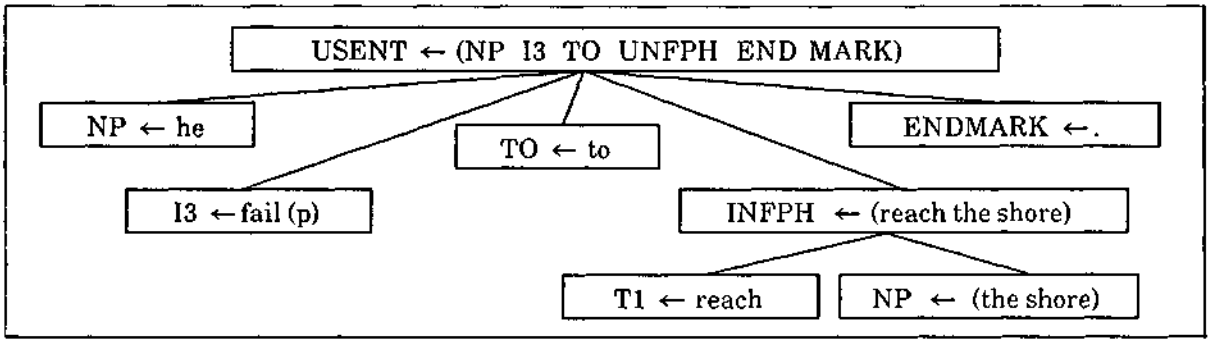

Fig. 4 Example of a simple sentence tree structure

There are many syntactic rules applied to simple sentence analyzing. Here, by inserting a rule

NP I3 TO INFPH ENDMARK → USENT

into the SCP listed above, the SCP for the example sentence is then given as

((RESTRICT +NP-1 NP-test)

(RESTRICT +I3 I3-test)

TO

(RESTRICT +INFinitive-PHrase INFPH-test)

>ENDMARK).

The analysis process unifies each variable in the SCP with each part of speech appearing in the sentence. For example, when unifying INFPH, an infinitive phrase must be verified as consisting of a transitive verb and an object noun. Each variable is bound to produce an SCP representation as an analysis result. If each unification succeeds, the simple sentence analysis is considered successful. At least one tree structure can be obtained as a result of the analysis, whose example is shown in Figure 4 for the sample sentence.

(3) Complex sentence analysis

An example of the syntactic rule for complex sentences is

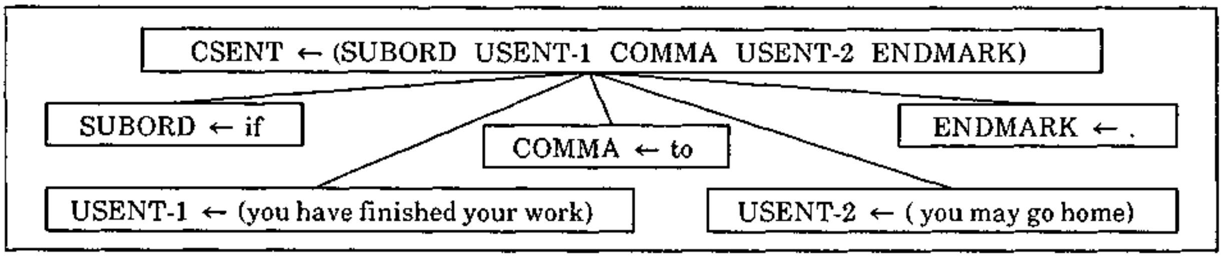

Fig. 5 Example of a complex sentence tree structure

SUBORD USENT-1 COMMA USENT-2 ENDMARK → CCENT

This syntactic rule represents a very common sentence consisting of a conjunction and a sentence as a subordinating clause having a comma in its end part and a complete sentence as a main clause. An example of the SCP representation for the sentence 'If you have finished your work, you may go home' is shown in Figure 5.

(4) Transfer

The Configuration Concoction Transfer utilizes Configuration Concoction Patterns (CCPs) for converting a source SCP representation into a target SCP representation. The number of the generated target SCP representations depends on the number of CCPs provided for the source SCP representation. The CCP involves configurational patterns of both languages, the source language and the target language, and combines a source SCP representation with a target SCP representation as in

((SUBORD USENT-1 COMMA USENT-2 ENDMARK)
(USENT-1 SUBORD COMMA USENT-2 ENDMARK)).

The representative framework for each configuration is the same as the SCP, thus each element or constituent is represented in the same framework. The CCP also involves the restrictions and functions necessary to produce a syntactically and semantically well-formed target SCPs. Therefore, again, semantic information can be encoded into those restrictions and functions, which check consistency when integrating the fragment SCPs into one larger SCP.

(5) Generation

Since the target SCP representation describes the configuration as a sequence of parts of speech respectively bound to words, the generation process task is to produce a sequence of terminal nodes after inducing the necessary word inflections, thus its process is called Pattern-driven Generation (PDG). However, there seems no need for further explanation about it since it is enough clear how to do it. The quality or naturalness of the generated translations depends on the information described in SCPs and CCPs.

6. Lexicon

Although LUTE-AID features a bi-lingual lexicon commonly used in analysis, transfer, and generation, it consists of seven sub-lexicons. The first is an isolated word lexicon. The second is one for phrase prepositions, each of which is manipulated as a preposition while actually consisting of several words including prepositions. Examples are 'on account of and 'at the bottom of'. The third is for phrase adverbs, each of which is treated as an adverb while having prepositional phrase configuration such as 'in the abstract' and 'by accident'. The fourth is for noun phrases whose roles are modifying nouns such as 'a bottle of' and 'a few'. The fifth is for verb idioms consisting of a combination for a verb and a preposition. The sixth is for compound nouns such as 'machine translation' and 'high school'. Finally, the seventh is for proper nouns such as personal names and city names.

The current LUTE-AID lexicon houses a large number of entries, especially for idiomatic expressions. Therefore, even if each lexical item contains little information, the entire lexicon can retain enough detailed information for effectively analyzing sentences and producing acceptable translations. Since there is a variety of approaches to automatic translation, providing a sufficient amount of lexical information is absolutely essential for the future extension.

7. Experiences

The examination was carried out in batch style, therefore neither interaction nor post-editing were performed. Grammatical information had been already encoded irrelevant with the corpus employed and the number of the SCPs was about 830 at that time. On the other hand, lexical entries were added for words appeared in the corpus but had not been involved in the entries. This was needed because the system was incapable of producing a translation such as '高等学校 *koutou gakkou'* whose equivalent is the compound noun, 'high school' (direct re-translation might be 'high-level school'. Therefore 'high school' was added as a compound noun entry in the lexicon. Otherwise the system would surely produce a translation like 'tall school'. The number of entries of the lexicon utilized in the examination was about 2,200, most of which were fundamental words. Additionally, about 150 entries for terminologies and proper nouns appeared in the corpus were added.

The test corpus was selected from two English language textbooks for Japanese students. Since one of the important objectives of textbooks is to teach as many sentence patterns in as short a time as possible, thus they constitute a very good corpus for the fundamental examination. One textbook is used for the second grade classes of secondary schools (hereafter, 'text 1') and the other is for the third grade classes (hereafter, 'text 2'). In Japan, students begin English class at the first grade of secondary school and finish it at the third grade in compulsory education. Therefore the corpus might has to include these all three textbooks to cover most fundamental sentence patterns, however, the textbook for the first grade involves just beginner's conversations such as 'How are you?', 'I am fine, thank you.', etc. or very fundamental sentences such as 'This is a pen.', therefore it's examination was omitted. Moreover, conversational, imperative, and interrogative sentences in the two text books were also removed from the corpus.

As the corpus, then, the text 1 involved 545 sentences and the text 2 involved 487 sentences. Among them, 378 translations from the text 1 (69%) and 236 translations from the text 2 (48%) were considered acceptable without any post-editing. The reason unacceptable translations were produced was almost entirely due to the lack of adequate SCPs or CCPs. According to the evaluation on the desk, it was recognized that most of them could be added while a small number of these patterns remained hard to encode without writing the translation directly by means of constant items. However, if such additional patterns without direct translations were incorporated, then 80% of the translations produced would be recognized as translations acceptable without post-editing in terms of the corpus actually employed in the examination.

References

[1] Nomura, H., Experimental Machine Translation Systems: LUTE, Second European-Japanese Workshop on Machine Translation, Dec. 1985.

[2] Nomura, H., Towards the High Ability Machine Translation, European-Japanese Workshop on Machine Translation, Nov. 1983.

[3] Saito, Y., Nomura, H., JMACS: Screen Editor for Japanese and English and Programs — A Kernel for Unified Research Environment, Proceedings of the International Conference of Text Processing with a Large Character Set, Oct. 1983.

[4] Nomura, H., Naito, S., Katagiri, Y., Shimazu, A., Translation by Understanding: A Machine Translation System LUTE, Proceedings of the International Conference on Computational linguistics, Aug. 1986. (This paper also contains a very brief description of RESOLUTE, while detailed descriptions have appeared in two other articles written in Japanese not listed here.)

[5] Nomura, H., Artificial Intelligence Approach to Machine Translation, ACM SIGART Newsletter, Jan. 1982.

[6] Shimazu, A., Naito, S., Nomura, H., Japanese Language Semantic Analyzer Based on an Extended Case Frame Model, Proceedings of the International Joint Conference on Artificial Intelligence, Aug. 1983.

[7] Iida, H., Ogura, K., Nomura, H., A Case Analysis Method Cooperating with ATNG and its Application to Machine Translation, Proceedings of the International Conference on Computational Linguistics, July 1984.

[8] Nomura, H., The LUTE Project, in Nirenburg, S. (ed), Special Section on Machine Translation of Natural Languages, ACM SIGART Newsletter, No.92, May 1985.

[9] Kudo, I., Nomura, H., Lexical-Functional Transfer: A Transfer Framework in a Machine Translation System Based on LFG, Proceedings of the International Conference on Computational Linguistics, Aug. 1986.

[10] Nomura, H., Modeling and Representative Framework for Linguistic and Non-linguistic Knowledge in Natural Language Understanding, Proceedings of the Germany-Japan Science Seminar, May 1986.