

Abstract.

IMT/EC is an English-Chinese machine translation system which integrates some outstanding features of the case grammar and semantic grammar into a uniform frame, uses various knowledge in the disambiguation, and tries to modify the object language by itself. In this paper, we first introduce IMT/EC's design motivation and overall architecture, then describe the design philosophy of its translation mechanisms and their processing algorithms.

1. The design motivation

The design of the IMT/EC system are motivated to develop new approaches to the English-Chinese machine translation, such as, to provide the system with powerful analysis mechanisms and MT knowledge base management system, as well as some exceptional processing and learning mechanisms, that is, to make the system be intelligent. In addition, it also tries to integrate as many advantages of conventional machine translation systems into a single system as possible, such as, to provide the system with powerful mechanisms for the processing of various ambiguities and contextual relations. The design of the IMT's translation mechanisms are based on the following considerations,

(1) SC-analysis

In the development of machine translation system, in order to disambiguate the source language, we have to analyze the input deeply to get the internal meaning representation of the source language. However, the deeper we analyze the input, the more we lose the clues about how to express the translation, also, that it results in extremely poor or no translations of sentences for which complete analyses can not be derived[Slocum 85]. To find a suitable analysis depth so as to get both clues about how to express the translation of the input and to disambiguate the input completely is almost impossible. In the IMT/EC, we try to design a simple grammar analysis mechanism -- SC-grammar analysis mechanism to inherit both the outstanding features of case grammar analysis and semantic grammar analysis so as to produce a high quality translation.

(2) Multi-language translations oriented

In present technical conditions, it is impossible to design a general internal meaning representation for all natural languages. Thus, the knowledge based multi-language oriented machine translation system is difficult to be marketed in the near future. A feasible way for designing multi-language oriented machine translation systems might be to separate the processing mechanisms from the language specific rules [as King et al. 85], that is, to apply the same processing mechanism with different language specific rules for different natural language pair translations. In the IMT/EC, we develop a general rule representation form for the representation of various knowledges used in the translation. Knowledges for different language pair translations are stored in the different packages of the knowledge base IMT-KB. The knowledge base are organized in multi-package and multi-level way so as to store rules for the translation of different language pairs and different phases of the processing. Thus, the system can be easily extended for multi-language translation purposes.

(3) Diversity processing

As the disambiguation rules are rather words specific, it is difficult to manage them in the same

way. To deal with this problem, we store these rules in their respected word entries and classify them as several categories in the IMT/EC. Each category corresponds to a general subroutine application mechanism, which apply the word specific rules and subroutines in the processing of translation. The subroutines are stored in a natural language specific subroutine package. Some word specific subroutines are directly stored in the respected word entry.

(4) Powerful exceptional processing

Since the natural language phenomena are so abundant that any existed machine translation system can not process all the phenomena, it is essential to provide an exceptional processing mechanism in the system to deal with exceptional phenomena. As IMT/EC incorporates some learning mechanisms, thus, it is more powerful in dealing with the exceptions than others.

(5) Automatic modification of the translation

Generally speaking, machine translation system can only produce rigid translations, it is a desire that MT systems be able to modify the output by itself so as to produce more fluent translations. IMT/EC tries to apply some common sense knowledge and linguistic knowledge of object language to disambiguate the input and modify the translations, thus, to improve the translation quality.

In the following paragraph, we focus on the translation procedure of the system and the algorithms related to it ignoring the knowledge base organization and management mechanisms.

2. The overall architecture of the system

The architecture of the IMT/EC system is as follow,

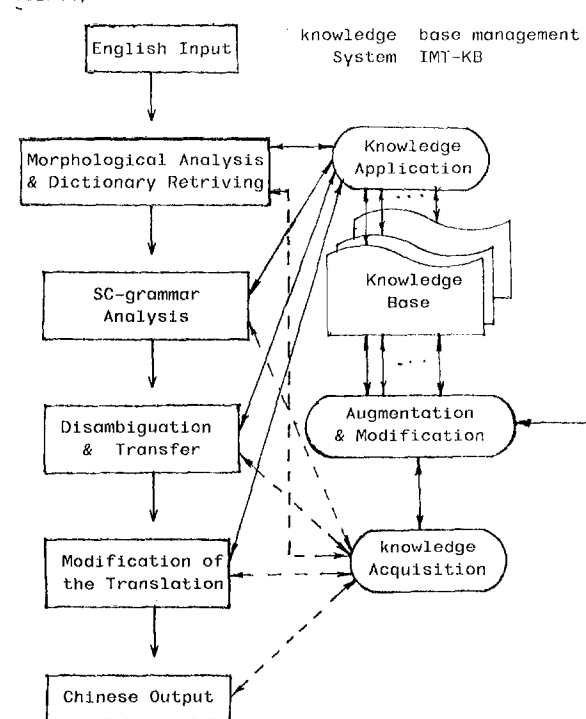


Fig. The architecture of the IMT/EC

As the rule base and dictionary in a machine translation system is so vast that it is impossible for human beings to find the confliction and implication among the rules. To modify a rule in the knowledge base often results in many side effects on other rules. Thus, it is necessary to provide a self re-organization and refinement mechanisms in the knowledge base.

In the IMT/EC, we design a special knowledge base management system IMT-KB to manage all the knowledge used in various processing phases of the translation. In addition, IMT/EC also provides a knowledge base augmentation and knowledge acquisition environment for the system to augment system performance by itself and for the users to improve the knowledge base.

The call relations connected by dotted lines in the figure above are executed only when the user sets the learning mechanisms in working status. These mechanisms can acquire new knowledge in the dynamic interactive, static interactive, or disconnected ways. They are primarily used to resolve the exceptional phenomena in the translation.

Dynamic Interactive Learning (DIL): Whenever the system encounters a sentence out of its processing range, it produces various possible translations for each segment of the sentence and interacts with human beings when necessary to select an appropriate translation of the segment and combine them to get a correct translation of the sentence. At the same time, it also creates some new rules to reflect the selections. That is, it learns some new knowledge.

Static Interactive Learning (SIL): Whenever the system encounters a sentence out of its processing range, it records down the sentence and its appearance context in a file. After the text has been translated, it begins to analyze the sentence in detail to get various possible translations for each segment of the sentence and interacts with human beings when necessary to get appropriate translations of the segments and combines them to get a correct translation of the sentence. At the same time, it also creates some new rules to reflect the selections, thus, to learn new knowledge.

Disconnected Learning (DL): Whenever the system encounters a sentence out of its processing range, it analyzes the sentence in detail to get all the possible translations, and then evaluates these translations according to the preference rules stored in the IMT-KB to select an appropriate translation and modify the related rules used in the analysis to reflect the selections. It skips over sentences which the translation can not be determined by the preference rules instead of interacting with human beings.

3. The translation procedure

IMT/EC's translation procedure is divided into several phases, i.e., morphology analysis and dictionary retrieving, SC-grammar analysis, disambiguation and transfer, modification of the translation etc.

The communications between translation mechanisms and the knowledge base are performed by the knowledge base management system IMT-KB, these operations includes getting a set of related rules and returning some information for the modification as well as augmentation of the MT knowledge base.

3.1. Morphology analysis and dictionary retrieving

In the IMT/EC, words in most common uses can be retrieved by either their base forms or their surface forms, while most of the other words can only be retrieved by their base forms. The tasks of the morphology analysis are to process the prefix, suffix, and compound words. Since these processings are completely natural language specific, in order for the processing mechanisms to be language independent, we develop a language independent morphology analysis mechanism to apply the language specific morphology rules in the morphology analysis.

The morphology analysis rule form is
 <surface pattern> -> <conditions> | <result>

Here,

<surface pattern> is the surface form of the word to be analyzed.

<conditions> is the application conditions of the rule.

<result> is the definition of the word base form analyzed.

For example,

```
(1) (* s)-> (verb *) | (def(*), SV)
(2) (* s)-> (noun *) | (def(*), PN)
(3)(*1 - *2)-> (word #1){word #2}|
                ((def(morphology #1),
                  def(morphology #2)), COM)
```

Here, *, *1 and *2 are variables indicating that it can be bounded to any sub-character string of the word to be analyzed, def(X) is the definition of X in the IMT-KB, SV, PN, COM are surface features of the word.

Rule (1) indicates that when the last character of the surface form of a word is 's' and the remained character string * in the word is a verb, then its surface feature is the singular verb form (SV) of the verb *. Thus, it returns the value of
 (def(*), SV)

as result.

Rule (2) indicates that when the last character of the surface form of a word is 's' and the remained character string * in the word is a noun, then its surface feature is the plural noun form (PN) of the noun *. Thus, it returns the value of
 (def(*), PN)

as result.

Rule (3) indicates that when the character string of a word comprises a character '-', the left part *1 and the right part *2 of '-' are both words, then it is a compound word of *1 and *2. Thus, it applies morphology rules to analyze the word *1 and *2, and returns the value of
 ((f(morphology #1),f(morphology #2)),COM)

as result.

Suppose that,

\$X indicates that X is a variable,

#X returns the character list of X,

&X returns the last character of X,

>X returns the first part of rule X or the first element of a list,

<X returns the remained part of X which
 (>X <X = X),

f(X,Y) returns the first different item pair of X and Y,

lookup(X) looks up the dictionary and returns the definition of the word X,

search(X) returns the morphology rules which includes character X,

check(X) tests whether two elements of the item pair X is unifiable or not,

null(x) tests whether list X is empty,

apply(g,X) returns the result of g(X),

t(X) tests whether result X needs further analysis and performs recursive analysis when necessary.

The algorithm for morphology analysis and dictionary retrieving is as follow.

INITIALIZE

\$X <- #word;

\$P <- search(& \$X);

\$P <- \$PU search(> \$X);

\$result <- ();

for \$rule e \$P do

{MATCH

\$PAT <- > \$rule;

\$COND <- >< \$rule;

\$RES <- << \$rule;

Loop

\$pair <- f(\$PAT, \$X);

if (null(\$pair)) goto TEST;

if (not(check(\$pair))) break;

\$PAT <- \$PAT o \$pair;

```

$PAIR_L ← $PAIR_L ∪ {$pair};
goto Loop;
TEST
for $COND1 ∈ $COND do
  ($PROP ← lookup(>w < ($COND1));
  if (not(apply(> $COND1, $PROP)) break;
  )
RESULT
$PROP ← lookup (> ($RES ∪ $PAIR_L));
$result ← $result ∪ (($RES ∪ $PAIR_L,
  $PROP));
)
if (pull($result)) return word
  else return t($result);
END.

```

3.2. SC-Grammar Analysis

The SC-grammar analysis mechanism of IMT/EC applies the SC-rules stored in the IMT-KB to disambiguate the structural ambiguities of the input sentences and produces the structural description for them. The grammar has some outstanding features of the case grammar and semantic grammar. The rule form is as follow,

```

<S-STRUCTURE> → <S-ENVIRONMENT> |
  <R-STRUCTURE>,
  <R-ENVIRONMENT>,
  <TRANSFER>.

```

Here,

<S-STRUCTURE> and <S-ENVIRONMENT> are rule conditions which defines the current structural form and contextual features of the input, <R-STRUCTURE> and <R-ENVIRONMENT> are result structural form and contextual features of the input, <TRANSFER> are the transformations related to the rule.

The structural forms, <S-STRUCTURE> and <R-STRUCTURE> are represented as strings of syntagmas and words. The contextual environments, <S-ENVIRONMENT> and <R-ENVIRONMENT> are represented as vectors, of which each element corresponds to an inter-sentential relation or a special case, their values are used to resolve the ellipsis, anaphora, tense and aspects etc. It is the principal contextual processing mechanisms in the IMT/EC.

Since the contextual vector is used only as a supplement to the pure semantic grammar analysis, especially in the processing of contextual relations, it is not necessary to analyze the input to the extent that one can get all the semantic relations of the input. Thus, the vector processing formalism is completely acceptable.

Two example rules are as follow,

```

NP VP → A | S, change(B1,X), INP IVP.
in NP → A1 | PP, change(B2,X), za1 |NP nuci.

```

SC-grammar analysis mechanisms receive the results of morphology analysis or previous SC-reduction, send the messages to the IMT-KB to get related rules, and apply these rules to reduce the input until a non-terminal symbol S is reduced, thus, to produce the structural description of the input.

The SC-grammar analysis algorithm of the system is:

(1) In the entries of the IMT-KB dictionary, we stored not only the word meanings and their disambiguation conditions, but also SC-phrase and semantic rules specific to the entry word. When analyzing a sentence, the system first retrieves the SC-phrase rules specific to the words appeared in the sentence, and applies these rules to find a list of possible phrases of the sentence from the context of the words in the sentence.

The phrase list returned is as follow,

```

X1(i1, j1)
X2(i2, j2)
.
.
.
Xn(in, jn).

```

Here, X₁, X₂, ..., X_n are phrase syntagma identifiers, i₁, i₂, ..., i_n and j₁, j₂, ..., j_n are ending positions of the phrases in the input sentence.

(2) Find a list of expectation paths from the phrase list as follow,

```

X1(p)(i1, j1) X2(p)(j1+1, k1) ... Xl(p)(m1+1, n1)
X1(q)(i1, j2) X2(q)(j2+1, k2) ... Xl(q)(m2+1, n2)

```

```

X1(p)(i1, j1) X2(p)(j1+1, k1) ... Xl(p)(m1+1, n1)
P(w1) P(w2+1) ... P(wl+1)

```

(Here, P(w) is the word w itself or its property, i is the current analysis position which initial value is 0, l is the expectation length defined by the user) and order them by means of the phrase ending positions n₁, n₂, ..., n_p from larger to smaller. These paths are used as heuristics in the analysis of the sentence. We try one new path at one backtracking.

(3) Send the analyzed component

$$M = Y_1(\dots) Y_2(\dots) \dots Y_l(\dots)$$

and the current expectation path to the IMT-KB to retrieve the SC-rules which head patterns contain a sub-string of

$$M \text{ Path} = \begin{cases} Y_1(\dots) \dots Y_l(\dots) X_1(\dots) \dots X_l(\dots) \\ \text{or} \\ Y_1(\dots) \dots Y_l(\dots) P(w_{i+1}) \dots P(w_{i+l+1}) \end{cases}$$

and organize these rules in a list according to their preferences from higher to lower. Then, it takes one rule from the rule list at one backtracking and go to (4) to apply the rule to reduce the input.

If no rule in the list can be successfully applied to reduce the input, the system gets the next expectation path from (2) and repeat (3). If all the expectation paths have been tried and no successful rule has been applied, it returns to the last analysis position to re-analyze the input. If the current analysis position is the beginning of the sentence, the system calls the exception processing mechanism to deal with this un-analyzable sentence.

(4) Match the rule head pattern with the current form of the input sentence. If there is a sub-pattern of the current sentence pattern that can match the rule head, then go to (5) else get the next rule from (3) and tries to re-match them.

(5) First, add some newly formed phrases into the phrase list in order for the backtracking of the analysis, then call the case analysis mechanism to check the current analysis results and the current form of the sentence to fill in the case frame A, B in the rule and the context vector. The case analysis algorithm is described in the following paragraph.

(6) Check A and the context vector to see whether their values are unifiable. If they are unifiable, then go to (7), else get the next rule from (3) and returns to (4).

(7) Store the backtracking information into the temporary stack, substitute the reducing part of the current sentence form with the reduced form, change the current analysis position to the last word of the newly reduced syntagma, and change the related element values of the context vector according to the element values of B.

If the current position is not the end of a sentence, then go to (2).

If the current position is the end of a sentence and the current form of the sentence is not S, then go to (2).

If the current position is the end of a sentence and the current form of the sentence is S, then go to (8).

(8) Call the semantic processing mechanism to check the result of the analysis to see whether it violates the English collocation rules. If the result violates the collocation rules, the system recovers to the status before the last reduction and gets the next rule from (3) to re-analyze the input. Otherwise, there will be two cases,

a. If the user only needs the most adequate translation, the system proceeds to analyze the next sentence.

b. If the user needs all possible translations, the system records down the current result and recovers to the status before the last reduction and gets

the next rule from (3) to re-analyze the input in order to get other analysis results.

As we have mentioned before, the case analysis in the SC-analysis is only a complement to the semantic analysis. It is mainly used to deal with the context relation and aspect, tense, modal etc. Thus, the system only needs to analyze those cases which can be used in those purposes. It is much simpler than the case analysis in the case grammar analysis.

The case analysis in the SC-analysis is performed by the following algorithm.

(1) Get the case expressions defined in the elements of vector A and B. The form of the element expressions of A and B is

$$S_j[i]:E$$

Here, $S_j[i]$ indicates the element case identifier (S_j) of the case frame A or B is corresponded to the case identifier $S[i]$ of the system case frame, i.e., system context vector. E is the expression used to get the value of the respected case.

(2) Retrieve the definition of the case identifiers from the system case frame and organize these case identifier into a list according to their preferences from higher to lower. The form is,

$$(S[1].subject:E1, S[12].object:E2, \dots, S[im].Em, \dots)$$

(3) Evaluate the value of the elements in the case identifier list, and fill them into the respected position in the case frame A and B. There are many cases in the evaluation.

- a. E is a constant, returns E,
- b. E is empty, evaluate the case value according the definition of the case identifier,
- c. If the case identifier is a syntagma identifier, then finds the value of the identifier from the analyzed input according to the heuristics provided by the expression E,
- d. If the case identifier is a semantic identifier, then call the semantic mechanism to get the value which can be filled into the case identifier from the input according to the heuristics provided by the expression E,
- e. For other case identifiers, call their respected subroutines to get the value of the case. These subroutines are defined by the rule designer.

The case analysis in the SC-analysis can solve the ellipsis, anaphora, and other contextual problems.

3.3. Semantic disambiguation and transformation

The SC-rules define not only the relations for the syntagma reduction, but also contextual vector value changes with respect to the reduction of a sentence, and the rules related transformations.

The transformation operation defined in the SC-rule is in the following forms,

$$IX \quad IX \quad \dots \quad IX$$

Here, IX, IX, ..., IX are translations of the syntagmas X, X, ..., X in the rule head. Their positions indicate the positions of the translations of the syntagmas. There will also be some indicators in the string which are used to indicate positions of the translations for inserting tense, voice, modal modifiers. These indicators are used as the heuristics of the semantic processing.

The transformation in the IMT/EC is relatively simple. It travels over the whole analysis tree from top to down, left to right, transfer every node when the node is traveled. The result of the transformation is the Chinese utterance of the sentence.

Rules with same head patterns may have different case frames A and B, in this case, they may correspond to different transformation operations. These rules are defined as two different rules by the rule designer. While in the IMT-KB, the system stores them as one rule with many candidate right patterns. Whenever the head pattern is successfully matched, the system sequentially checks these candidates until one of them is satisfied and records down the current successful position so that backtracking mechanism can

get the other candidates when necessary.

The tasks of the semantic processing in the IMT/EC are to check the results of the analysis to see whether they satisfy the syntax or semantic collocation rules defined in the IMT-KB, to produce the suitable modifiers for expressing the tense, voice, aspects and so on in the Chinese. In some cases, it also apply the well formed world knowledge defined in the IMT-KB to eliminate some illegal expressions and extend the meanings of some ambiguity words.

Since the SC-analysis is based on the semantic grammar analysis, most of the syntax and semantic ambiguities are solved in the reduction operations. Even though the case analysis in SC-analysis is aimed mainly to resolve the contextual problems, they can also solve some ambiguities among a sentence. That is, the semantic processing in the IMT/EC is oriented to specific ambiguities and inter-sentential case value evaluations. Though the processings are different in different phases of the translation, they can be categorized as,

(1) determining the value of a specific semantic identifier, such as time adverbial, place adverbial, anaphora etc.

When a specific semantic identifier is concerned, the semantic processing mechanisms first finds the key word which can match the semantic identifier from the sentence, such as word with time, place properties, then get the phrase which comprises the key word in the sentence, and return the phrase as the value of the identifier.

Only simple anaphora phenomena are considered in the IMT/EC. They are processed in two different ways. One is to compare the synonyms to find the anaphora words, the other is to find the suitable anaphora content through the position relations, such as, in some specific context the word 'which' can refer to the noun phrases immediately before it.

(2) checking the collocation of syntagmas.

There are three possible categories of collocation in the analysis results,

- <1> X W \rightarrow (W \Rightarrow C₁)
- <2> W Y \rightarrow (W \Rightarrow C₂)
- <3> X W Y \rightarrow (W \Rightarrow C₃)

Here, X, Y may be strings of words or syntagmas, W is a specific word. The above expressions means that,

- <1> W appears after string X and functions as speech C₁,
- <2> W appears before string Y and functions as speech C₂,
- <3> W appears between string X and Y and functions as speech C₃.

The related word definition in the IMT-KB dictionary is as follow,

$$W := C_1 \begin{pmatrix} (E_1 \Rightarrow M_1) \\ (E_2 \Rightarrow M_2) \\ C_2 \end{pmatrix} \begin{pmatrix} (E_3 \Rightarrow M_3) \\ \vdots \\ C_m \end{pmatrix} (E_n \Rightarrow M_n)$$

Here, C is the speech category, E is the context structure of word W, M is the meaning of word W.

The semantic processing mechanism retrieves the collocation rules specific to words of the sentence from the IMT-KB, and applies these rules to check the analysis result to see whether there is any violation between the analysis result and collocation rules. If there is, returns fail.

(3) checking the distant contextual relations.

There are also three possible categories of distant contextual relations appeared in a sentence,

- X ... W [m] \rightarrow (W \Rightarrow C₁)
- W ... Y [n] \rightarrow (W \Rightarrow C₂)
- X ... W ... Y [m, n] \rightarrow (W \Rightarrow C₃)

Here, X, Y, W, C have the same meanings as in the (2). n, m are optional, they defines the relative position between the word W and X/Y. When n, m = \emptyset , they are the cases described in (2). When n, m is not

defined, they indicate any position before/after the word W. These distant contextual relation rules are defined in the IMT-KB in the same way as in (2).

If m and/or n are present, the semantic processing mechanism finds m/n word before/after the word W in the sentence, and tries to reduce that word and its adjacent words into X or Y. If they can be reduced, and the word W functions as the same category as defined in the rule, then succeeds, also eliminates the analysis.

If m and n are not defined, then try to find the word before/after the word W which can be reduced into X or Y together with its adjacent words. If there are no such element in the sentence, then returns fail.

(4) creating Chinese modifiers to express the tense, voice, modal and so on and insert these modifiers in the translation according to the position mark appeared in the rule.

The processing procedure is as follow,

- a. Get the marks of the tense, voice, modal etc.
- b. Call the corresponding subroutines defined by the rule designer to determine an appropriate modifier for the mark. This is based mainly on the contextual structure of the analysis result.
- c. Insert the modifier in the position of the translation marked by the marker.

For example, if a verb is in the '-ing' form and there is no time adverbial in the input, the tense of the context are all progressive, then ignores the time mark. If the predicate are 'be going to', then translates it as 'dashuang' ignoring the time mark.

The world knowledge rules are defined in the same form as the semantic rules. The application of these rules are to test the context to find the semantic features of the situation and compare these to the world model definition defined in the world knowledge rule to determine the situation of the utterance, and then determine the correct translation or extend the meanings of related words.

Every semantic processing mechanism mentioned above corresponds to a specific processing subroutine. These subroutines are called in the grammar analysis and transformation processing to perform the related semantic processing. (1) is primarily used in the case analysis, (2) and (3) are primarily used in checking the analysis result and disambiguations in the analysis and transformation, (4) is primarily used in the transformation.

The grammar and word transformation algorithm is,

- (1) Current-node ← root of the analysis tree,
- (2) If the current-node is a leaf node, go (4),
- (3) The current-node is not a leaf node, the processing are as follow,
 - a. if all the elements in the transformation expression of the node are constant, go (5),
 - b. if all the variables in the transformation expression of the node are substituted by constants, then call semantic processing mechanism to create suitable modifiers. Go (5).
 - c. if there are some unsubstituted variables in the expression of the node, set these variables as current-node one by one, and uses the results returned by each subnode to replace the variables.
- (4) When the current-node is a leaf node, that is, it is a specific word or an idiom, then retrieves its definition from the IMT-KB, call the semantic processing mechanism to determine an appropriate meaning for it according to the tree structure.
- (5) If the current-node is root node, then returns the current form of the transformation expression as the translation of the sentence. Otherwise, returns the expression to the parent node, recovers the parent node as current node. Go (2).

3.4. The modification of the translation

The objective of the automatic modification of the translation is to improve the readability of the translation, but this sacrifices part of the accuracy. It is more suitable for the non-scientific literature

translation.

The main tasks comprises:

- a. Change the order of the phrases and words of the translation,
- b. Substitute some words which collocation is not commonly used in the Chinese utterance for the synonymous words,
- c. Insert some conjunctive words when necessary,
- d. Eliminate some redundant words.

The algorithm for these processing is,

(1) According to the Chinese collocation rules defined in the IMT-KB, changes the words and phrases order of the translation which are not in accord with the collocation conventions in Chinese, such as,

Budan ... , Erchia ...

(2) According to the co-occurrence rules of the Chinese words defined in the IMT-KB, check the uses of the Chinese words in the translation. If they are not in accord with the co-occurrence rules, then replaces these words with the Chinese synonymous words until they are accord to the rules. If there is no suitable synonyms, then tries to extend the meaning of some words. The meaning extending rules are defined in the word entries. Its form is as follow,

```
<word> :- <condition 1> | <extension 1>
          <condition 2> | <extension 2>
          .
          .
          <condition n> | <extension n>
```

Here, <word> indicates the word appeared in the sentence, <condition> defines the extending conditions, <extension> is the utterances extended.

If the word can not be replaced or extended, then just returns the source translation.

(3) Check the translation to find the redundant words and eliminates them. The form of deletion rule is,

$X Y X Z \rightarrow p(X), p(Y) | X Y Z$

such as, 'NP de NP de → NP NP de'.

Since the modification has no absolute standard and requires a large amount of world knowledge, it is rather difficult to solve this problem in one day. In the IMT/EC, we only deal with the most simple cases. More complex situations can be solved with the application and improvement of the system. Thus, the system is designed to be easily extended with the application.

If the user needs high quality translation, he may call the post editing subroutine to modify the translation by human beings or with the aid of human beings. At the same time, we can also set the learning mechanisms in working status to trace the modification procedure of human beings and produce some useful rules for the system.

4. Summary

In conclusion, we have introduced the translation processing procedure of the English-Chinese machine translation system IMT/EC, and describe its principal processing algorithms.

Acknowledgement: We would like to thank Hong Xiong, Zhang Yujie, Ye Yimin, Tong Jiaxian, Zong Liyi, Zhang Zifa, Chen Zizong, Chen Zizeng and Fu Wei for their cooperation in the implementation of IMT/EC.

5. Reference

- [1] Axelbiewer, Christian Fennyrol, Johannes Ritzke, Erwin Stegentritt (1985), ASCOF-A modular multilevel system for French-German translation, CL, Vol.11, No. 2-3, p137-154, 1985.
- [2] Bernard Vauquois and Christian Boitet, Automated

translation at Grenoble university, CL, Vol.11, No.1, p28-36, 1985.

[3] Bozena Hennisz-Dostert et al., Machine Translation, Mouton publishers, Hague, Paris, New York, 1979.

[4] Harry Tennant, Natural Language processing, Pietrocella books, New York, 1981.

[5] Hiroshi Uchida, Fujitsu machine translation system: ATLAS, FGCS, Vol.2, No.2, p95-100, 1986.

[6] Jaime G. Carbonell and Masaru Tomita, New approaches to Machine Translation, TR-CMU-CS-85-143, Carnegie-Mellon University, 1985.

[7] Jonathan Slocum, A survey of machine translation: its history, current status and future perspectives, CL, Vol. 11, No. 1, p1-17, 1985.

[8] Kazunori Muraki, VENUS: Two-phrase machine translation system, FGCS, Vol.2, p121-124, 1986.

[9] Martin Kay, The MIND system, Natural Language processing, edited by Rustin, Algorithmics press, New York, 1973, p155-189.

[10] Makoto Nagao, Current Status and future trends in machine translation, FGCS, Vol. 2, No. 2, p77-82, 1986.

[11] M.Nagao, J.Tsujii and J.Nakamura, Science and Technology agency's machine translation project, FGCS, Vol.2, No.2, p125-140, 1986.

[12] Muriel Vasconcellos and Marjorie Leon, SPANAM and ENGSPAN: machine translation at the PAN American health organization, CL, Vol.11, No.2-3, p122-136, 1985.

[13] Paul L.Garvin(ed.), Natural Language and the Computer, McGraw-Hill book, New York, London, 1979.

[14] Pereira F. and Warren D., Definite clause grammar for language analysis, Artificial Intelligence, Vol.13, p231-278, 1980.

[15] Pierre Isabelle and Laurent Bourbeau, TAUM-AVIATION: its technical features and some experimental results, CL, Vol.11, No. 1, p18-27, 1985.

[16] Richard E. Cullingford, Word-meaning selection in multiprocess language understanding, IEEE Trans. on Pattern analysis and machine intelligence, Vol. PAMI-6, No.4., July 1984, p493-509.

[17] R.F.Simmons, Technologies for machine translation, FGCS, Vol.2, No.2, p83-94, 1986.

[18] Rod Johnson, Maghi King, and Louis des Tombe, EUROTRA: A multilingual system under development, CL, Vol.11, No.2-3, p155-169, 1985.

[19] Roger Schank, The conceptual analysis of natural language, Natural Language processing, edited by Randall Rustin, Algorithmics Press, New York, 1973, p291-311.

[20] Bozena Hennisz-Dostert, R.Ross Macdonald and Michael Zarechnak, Machine translation, Mouton Publisher, Hague, Paris, New York, 1979.

[21] S.Amano, The Toshiba machine translation system, FGCS, Vol.2, No.2, p121-124, 1986.

[22] Terry Winograd, Language as a Cognitive(Vol 1) process, Addison-Wesley Publishing Company, California, London, 1983.

[23] Vinfield S.Bennett and Jonathan Slocum, The LRC machine translation system, CL, Vol.11, No.2-3, p111-121, 1985.

[24] V. Wilks, The Stanford machine translation project, Natural Language Processing, edited by Randall Rustin, Algorithmics Press, New York, 1973, p243-291.

[25] Yoshihiko Nitta, Problems of machine translation systems-effect of cultural differences on sentence structure, FGCS, Vol.2, No.2, p117-120, 1986.