An English-Korean System for Human Assisted Language Translation

Seung-Sik Kang, Kwang-Seob Shim, Byoung-Tak Zhang
Hyuk-Chul Kwon, Chi-Su Wu, and Yung-Taek Kim

Dept. of Computer Engineering
Seoul National University
Seoul 151, Republic of Korea

ABSTRACT

KSHALT (English-Korean System for Human Assisted Language Translation) is a machine translation system from English to Korean based on transfer approach. It consists of four phases such as English parsing, English analysis, English-Korean transfer and Korean generation. Each phase is designed to have modularity and extensibility and driven by the information from various dictionaries for more efficiency.

1. INTRODUCTION

In machine translation the accuracy of translation usually depends on the range of domain. Presently IBM system manual is the range of this project, and the accuracy of translation is being improved toward the level of satisfaction.

For English parsing, KSHALT uses PEG (PLNLP English Grammar) which is developed at IBM Watson Research Center. And the output of PEG is analyzed to be transformed into correct form. The structure of KSHALT is shown in Fig.1.

During the English analysis phase, tree selection procedure for multiple tree output is activated and a standardized English intermediate representation is constructed through the analysis. English-Korean transfer phase consists of lexical substitution of English word by Korean word and word order conversion from English sentence to Korean sentence is decided. Korean generation phase receives a

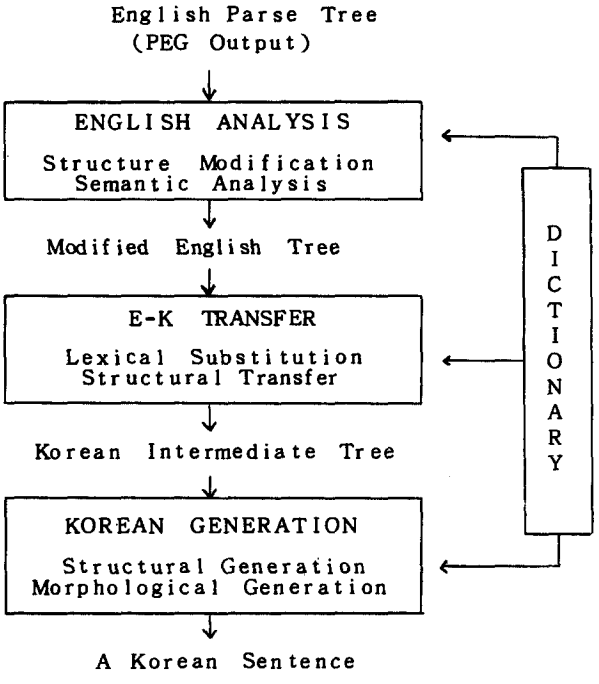

Fig.1 Structure of KSHALT

Korean intermediate representation as input and transforms it into a Korean representation with proper word order. Then Korean sentence will be generated through the morphological generation of Korean grammar. The implementation of each phase is shown through the formulations of rules.

2. ENGLISH ANALYSIS

In this phase, English parse tree is analyzed and modified to obtain correct tree. The detailed works, such as tree selection,

CH2423-2/87/0000-0509 $1.00 © 1987 IEEE

structure modification, processing of modal information, correction of prepositional phrase and the usage of clauses, are treated

2.1 Tree Selection

English parser often produces multiple trees for a given input sentence. The reasons for multiple trees are analyzed as follows.

1) Some word can be interpreted as two different part-of-speech, e.g. "like" as preposition or verb.
2) A word or a phrase can describe two different words in a sentence, e.g. in "User only need to ···", "only" can be attached to "user" or "need".
3) The syntactic structure of a sentence can be interpreted as two or more tree forms, e.g. "both SNA and non-SNA networks" can generate two different trees.
4) etc.

The algorithm of selection tree is defined as follows.

1) Compare first two trees to identify the difference.
2) Apply selection rules for the difference.
3) For more than two multiple trees repeat 1 and 2.

The word "like" generates two different trees, and the selection rule is applied as shown in Fig.2.

```
(LIKE
  if
   (has-word "like")
   ((lex-cat "like" xboth)
    is
   (PREP or VERB*))
  then
   (select-tree
     ((lex-cat "like" which) is PREP)))
```

Fig.2 Example of selection rule

Some trees which can't be handled by the selection rule are to be handled by human assistance.

Selection rules are formulated as shown in Fig.3.

```
⟨sel-rule⟩
   ::=(⟨rule-name⟩
       IF {⟨cond⟩}* THEN ⟨action⟩)
⟨rule-name⟩ ::= ⟨symbol⟩
⟨cond⟩      ::= ⟨lisp-ftn⟩ | ⟨modi-ftn⟩
⟨action⟩    ::= (SELECT-TREE ⟨cond⟩)
⟨lisp-ftn⟩  ::= (⟨lftn-name⟩ ⟨args⟩)
⟨modi-ftn⟩
   ::= (⟨test-ftn⟩ IS ⟨atomics⟩)
⟨test-ftn⟩
   ::= (⟨tftn-name⟩ ⟨args⟩ ⟨test-args⟩)
⟨lftn-name⟩ : :=  ⟨symbol⟩
⟨tftn-name⟩
   ::= HAS-WORD | LEX-CAT | ···
⟨args⟩
   ::= {⟨lisp-atom⟩ | ⟨lisp-ftn⟩}*
⟨test-args⟩ ::= BOTH | XBOTH | WHICH
⟨atomics⟩
   ::= ⟨lisp-atom⟩
       | (⟨lisp-atom⟩ OR ⟨lisp-atom⟩
⟨lisp-atom⟩
   ::= ⟨symbol⟩ | ⟨string⟩ | ⟨integer⟩
```

Fig.3 Formulation of Selection Rule

2.2 Structure Modification

Usually predicate parts of English parse tree are not grouped into a verb phrase and object parts in prepositional phrases are not grouped into a noun phrase. So, the range of predicate parts and object parts are not specified clearly as in (1).

```
(1) (DECL (NP      ·····)
          (VERB*   ·····)
          (NP      ·····))
```

In this case new VP node is constructed as in (2).

```
(2) (DECL (NP      ·····)
          (VP(VERB* ···)
             (NP    ···)))
```

New VP and NP nodes are constructed for the following cases.

1) Predicate part in a main clause.
2) Predicate part in a compound sentence.
3) Predicate parts in infinitive clauses, participle clauses and abbreviated clauses.
4) Object parts in prepositional phrases.

2.3 Modal Information

In English, some pieces of modal information(auxiliary verb, negation) are syntactically analyzed for a verb as in (3).

```
(3) (VERB "could" ("can" PS))
    (AVP  (ADV* "not" ("not" BS)))
    (VERB "be" ("be" PS))
    (VERB* "used" ("use" PS))
```

But in Korean, they appear as a postpositional form for a verb node. Thus the modal information is to be attached to a corresponding verb node as in (4).

```
(4) (VERB* "use" (TENSE PAST)
       (VOICE PASSIVE) (AUX CAN)
       (NEGATION  T)(PROGR NIL))
```

2.4 Correction of Prepositional Phrase

PEG has a rule that prepositional phrase is attached to the very left positioned part-of-speech. Prepositional phrase in (5) describes "data" but it must be corrected to describe "store" as in (6).

```
(5) (VP (VERB* "store" PS)
        (NP (NOUN* "data" (SG PL))
            (PP (PREP "in")
                (NP (NOUN* ···)))))

(6) (VP (VERB* "store" PS)
        (NP (NOUN* "data" (SG PL)))
        (PP (PREP "in")
            (NP (NOUN* ···))))
```

For the correction of structure, the prepositions are classified and utilized according to their properties classified as in the following.

1) Prepositions which describe NP: of, like, between, under, ……
2) Prepositions which describe VP: into, during, without, ……
3) Prepositions which describe both NP and VP: on, in, to, with, ……

2.5 Usages of Infinitive Clauses and Participle Clauses

The usages of infinitive clauses and participle clauses are defined in this section.

The usage of an infinitive clause is classified into ADJECTIVE, ADVERB, NOUN. And the usage of a participle clause is classified into NOUN, PURPOSE, AS. Rules for usages are as follows.

1) Infinitive clause which follows "how" is used as NOUN.
2) Participle clause which describes a verb phrase is used as ADVERB.
3) etc.

Semantic information is supplied from the dictionary for higher quality of classification.

3. E-K TRANSFER

In this phase the Korean intermediate representation is generated from the English parse tree. This phase consists of three subphases, such as pretransfer, lexical substitution and structural transfer.

3.1 Pretransfer

It is necessary to reduce the structural difference between the two languages, especially between the two different families of language. For example, there is no syntactic structure of a Korean sentence like (1). And the sentence (2) is much closer to Korean structure. Therefore, the sentence (1) is to be transformed into the sentence (2).

(1) "It is important for the user to specify the file."

(2) "That the user specify the file is important."

For this purpose, we introduce Pretransfer before Lexical Substitution. The pretransfer is carried out by the application of subtree-to-subtree rules, as shown in the Fig.4.

```
(
   GROUP IS <symbol>
```

```
  [RULE ID IS <symbol>]
  [PARENT IS <symbol>]
  [ITERATION IS <iteration-type>]
  PATTERN <match-pattern>
  [CONDITION {<function-call>}*]
  [PRE-ACTION{<function-call>}*]
  CREATE <create-pattern>
  [POST-ACTION {<function-call>}*]
)

<iteration-type> ::= ALLOWED |
                     NOT ALLOWED
<match-pattern>  ::=
  list of <match-element>
<create-pattern> ::=
  list of <create-element>
<match-element>  ::= ∂ |
                     <var> |
                     <symbol> |
                     <string> |
                     <match-pattern>
<create-element> ::= NIL |
                     <var> |
                     <symbol> |
                     <string> |
                     <create-pattern>
```

Fig.4 Formulation for transfer

Fig.5 is an instance of pretransfer rule that transform sentence (1) to sentence (2).

```
( GROUP IS PRE
  RULE ID IS PRE-SENT-0001
  PARENT IS SENT
  PATTERN
   (*1* (NP (NOUN "it" ∂))
        (VP (VERB "be" *2*)
            (AJP *3*)
            (SENT (TYPE INFCL)
                  (FUNC ∂)
                  *4*)))
  CREATE
   (SENT *1* (NP (SENT (TYPE COMPST)
                       (FUNC "that")
                       *4*))
             (VP (VERB "be" *2*)
                 (AJP *3*))) )
```

Fig.5 A pretransfer rule for transforming sentence (1) to sentence (2)

Other works are also done for later subphase. For example, omitted noun phrase within relative clause should be restored for lexical substitution subphase.

3.2 Lexical Substitution

At this subphase the English terminal words are substituted by proper Korean words Some informations are added to the parse tree, and they are referred at structural transfer subphase.

Since a word can be used by multiple meaning, it is difficult to select a proper Korean word from many alternatives. In particular, for the case of preposition, the complexity increases since it brings the variety of contexts.

There are other problems which arise from the difference of English and Korean lexical usages. For example, some transitive verbs in English are substituted by intransitive verbs of Korean, and the same with the reverse case, as shown in (7).

(7) "A user enters the CMS/DOS environment."

→ "사용자가 CMS/DOS환경 으로 들어간다."

The words are not always substituted in one-to-one manner. One-to-multiple or multiple-to-one substitutions are possible. Example (8) will show that an English word 'any' brings two Korean words, '어떤' and '라도', and two English words, 'make' and 'change' do one Korean word '변경하다'. The verb conjugation of '변경하다' to '변경할수 있다' is done at the generation phase.

(8) "Any user can make change to the file contents."

→ "어떤 사용자라도 화일의 내용을 변경할 수 있다."

By restricting the domain of translation into a specific field, we can partially resolve this difficulty. Currently we restrict the domain to IBM computer manuals. The dictionary provides the necessary informations for the cases.

Insertion of new phrases into a sentence or deletion of some phrases from a sentence

should be specified in a dictionary as in the case of (8), where '라도' is inserted because of the word 'any' which is substituted into '어떤'. The sample entries of the dictionary are shown in Fig.6.

```
(VERB "enter"
      ((DO ("로" "") (G 1))
       ("들어가" "")) …)
(VERB "make"
      ((DO (H "change") (CUT))
       (PP * (H ("to" "을" "" (G 3))))
       ("변경하" "여" )) … )
(ADJ "any"
     ((/NP (POST "" "든지"))
      ("어떤" )) … )
```

Fig.6 Sample entries of dictionary

(G 1) and (G 3) in Fig.6 specify that postpositions '로' and '을' belong to group 1 and group 3 of adverbs respectively. These informations are utilized at structural transfer step.

3.3 Structural Transfer

After the lexical substitution subphase, the terminal words are substituted by proper Korean words. And the tree is transformed into a Korean intermediate representation. After this transformation, the tree representation will construct meanings to a certain degree. The morphological processing such as verb conjugation and postposition adjustment is done at the Korean Generation phase to construct a complete meaning.

The transformation uses structural transfer rules that have the same formulation with that of pretransfer. If a sentence contains several adverbs, postpositions and adverbial endings, then the relative order between the adverbs could not be determined only by rule application. The naturalness of a sentence depends heavily on the relative order of adverbs. To solve this problem, we classify adverbs into several groups, and assign predetermined precedence to each group. The dictionary provides such information. For example, if a sentence has two adverbs, one belongs to group i and the other to group j, and i> j, then the adverb of group j would follow that of group i according to the precedence.

4. KOREAN GENERATION

In this phase Korean sentences are generated from the intermediate representation which is the output of the transfer phase. This phase consists of structural generation and morphological generation.

4.1 Generation Rules

The structural and morphological generations are rule-based and each rule is of the form of (if ⟨condition⟩ then ⟨action⟩). The rules are classified into several rule classes, depending on the usage category of rules. Each rule class is formulated as in Fig.7.

```
⟨rule-class⟩ ::=
      (⟨classname⟩
       ⟨rule⟩*
       [ (default ⟨action⟩) ] )
⟨rule⟩       ::= (if ⟨condition⟩
                     then ⟨action⟩)
⟨condition⟩  ::= ⟨function⟩
⟨action⟩     ::= ⟨function⟩
⟨function⟩   ::= (functor ⟨args⟩)
⟨args⟩       ::= ⟨arg⟩*
⟨arg⟩        ::= ⟨function⟩ | ⟨atom⟩
⟨atom⟩       ::= ⟨lisp-atom⟩
⟨functor⟩    ::= PVOWEL | ERASE | GEN…
⟨classname⟩  ::= QUNIT | PAST …
```

Fig.7 Formulation for Korean Generation

The rule class SUBJ-POST contains four rules for each Korean subjective postposition, such as "은", "는", "이", and "가". For the case of "은" the rule has the form:

```
(if (cond-1,, cond-n are satisfied)
 then (generate "은")).
```

4.2 Structural Generation

Intermediate representation is adjusted for more natural Korean sentence. The works for this phase are categorized into the following:

(1) Addition of a Korean quantifier
(2) Elimination of duplicate subject
(3) Relocation of a Korean word
(4) Insertion of a punctuation mark
(5) Reordering of the word order etc.

For the category (1), by applying the rules of the class QUNIT "세 가지의 단계" is generated for the English phrase "three steps" as follows.

```
(QUNIT
   (if (NOUNIS " 단계 ")
    then (INSERT '(ADJ " 가지의 ")))
   (if (NOUNIS " 책 " )
    then (INSERT '(ADJ " 권의 ")))
        ... ...
   (default (INSERT '(ADJ " 개의 "))).
```

4.3 Morphological Generation

In this phase lexical and grammatical morphemes of Korean are generated and combined into a sentence. The specific morphemes generated in this step are :

(1) Auxiliary-stems
(2) Endings for predicates
(3) Postpositions etc.

The detailed works for morphological generation are shown in the following example.

Example: Korean generation for the English phrase "could not be used"

(1) Intermediate representation

```
(VERB
  ("사용하"  " 여 " )
  ("use"  (VOICE PASSIVE) (AUX CAN)
          (NEGATION T) (TENSE PAST)
          (PROGR NIL) (TYPE DECL)))
```

(2) Rule application order

(VOICE AUX NEGATION TENSE TYPE).

(3) Application of rules

```
" 사용하 "
   ↓ ..... VOICE (PASSIVE)
" 사용되 "
   ↓ ..... AUX (CAN)
" 사용될 수 있 "
   ↓ ..... NEGATION
" 사용될 수 없 "
   ↓ ..... TENSE (PAST)
" 사용될 수 없었 "
   ↓ ..... TYPE (DECL)
" 사용될 수 없었다."
```

(4) Rules used for the step PAST

```
(PAST
 (if (and (CAT 'VERB) (CONJUG "ㄷ")
          (PVOWEL *stem*))
  then (and (ERASE "ㄷ")
            (GEN "ㄹ았")))

 ...
 (if (and (CAT 'VERB) (CONJUG "여"))
  then (GEN "였"))

 ...
 (if (NVOWEL *stem*)
  then (GEN "었"))).
```

5. CONCLUSION

Some results of translations are shown in appendix. The domain and object can be read through the appendix. Since PEG is developed for unlimited range of English sentence and unrestricted object, KSHALT also can be used for wider range and other purposes. Since dictionary is affecting directly the accuracy of each phase, the organization and the size of dictionary will become the major factor of the project.

REFERENCES

[1] Y.H. Kim, "A Study of Case Grammar in Korean Language", M.S. Thesis, Yonsei Univ., 1973.

[2] T. Tsutsumi, "A Prototype English-Japanese Machine Translation System for Translating IBM computer manuals", Proceedings of Coling 86, 1986

[3] M. Nagao, J. Tsujii, "The Transfer

Phase of the Mu Machine Translation System", Proceedings of Coling 86, 1986.

[4] J. Slocum, "A Survey of Machine Translation : its History, Current Status, and Future Prospects", AJCL 11-1,1985.

[5] A. Biewer, C. Feneyrol, J. Ritzke, E. Stegentritt, "ASCOF-- A Modular Multilevel System for French-Genrman Translation", Computational Linguistics, Vol.11, 2-3, pp.91-110, 1985.

[6] M. Harada, "An English-to-Japanese Machine Translation System SHALT - Japanese transformation", Science Institute, IBM Japan.

[7] G. Gazdar, G.Pullum, Generalized Phrase Structure Grammar: A Theoretical Synopsis, Indiana University Linguistics Club, reading, 1984.

APPENDIX : KSHALT outputs

The system gives a message with the current time and date.

시스템은 현재의 시간과 날짜에 대한 메세지를 전한다.

Financial analysts can create models to project investment opportunities.

재정 분석가는 투자 기회를 계획하기 위하여 모델을 생성할 수 있다.

CMS commands let the terminal user process CMS data stored on DASD.

CMS 명령어는 단말기 사용자가 DASD에 저장되는 CMS 데이타를 처리하도록 한다.

The Control Program (CP) manages the resources of the system and creates virtual machines where operating systems can run.

제어 프로그램(CP)은 시스템의 자원을 관리하고 운영 체제가 실행될 수 있는 가상 기계를생성한다.

Inexperienced professionals and non-professionals can quickly learn how to use VM/SP for their personal needs.

경험이 부족한 전문가와 비전문가는 그들의 개인적인 필요를 위하여 VM/SP을 사용하는 방법을 빨리 배울 수 있다.

System operators and system programmers can receive hands-on training by using a virtual machine rather than using real system time.

시스템 조작자와 시스템 프로그래머는 실 시스템 시간을 사용하는 대신 가상 기계를 사용하는 것에 의해 직접적인 훈련을 받을 수 있다.

These permanent virtual disks are assigned to a virtual machine each time a user starts a terminal session with a virtual machine.

이 영구적인 가상 디스크는 사용자가 가상 기계로 단말기 세션을 시작할 때마다 가상 기계에 할당된다.

When a user enters the CMS/DOS environment, he or she can use available comands to develop and test DOS programs.

사용자는 CMS/DOS 환경으로 들어갈 때 DOS 프로그램을 개발하고 조사하기 위하여 유용한 명령어를 사용할 수 있다.