*Computational Intelligence*, Volume 18, Number 3, 2002

# GENERATE AND REPAIR MACHINE TRANSLATION

KANLAYA NARUEDOMKUL
*Department of Mathematics, Mahidol University, Bangkok, Thailand*

NICK CERCONE
*Computer Science Department, University of Waterloo, Waterloo, Ontario, Canada*

We propose Generate and Repair Machine Translation (GRMT), a constraint-based approach to machine translation that focuses on accurate translation output. GRMT performs the translation by generating a *Translation Candidate* (TC), verifying the syntax and semantics of the TC and repairing the TC when required. GRMT comprises three modules: Analysis Lite Machine Translation (ALMT), Translation Candidate Evaluation (TCE) and Repair and Iterate (RI). The key features of GRMT are simplicity, modularity, extendibility, and multilinguality.

An English–Thai translation system has been implemented to illustrate the performance of GRMT. The system has been developed and run under SWI-Prolog 3.2.8. The English and Thai grammars have been developed based on Head-Driven Phrase Structure Grammar (HPSG) and implemented on the Attribute Logic Engine (ALE). GRMT was tested to generate the translations for a number of sentences/phrases. Examples are provided throughout the article to illustrate how GRMT performs the translation process.

*Key words*: Generate and Repair Machine Translation, machine translation, multilingual machine translation.

## 1. INTRODUCTION

In 1996, we proposed Generate and Repair Machine Translation (GRMT) (Naruedomkul and Cercone 1997). GRMT is designed to increase the accuracy and efficiency of machine translation (MT). The GRMT architecture is designed to take advantage of, and have advantage over, the direct, transfer, interlingual, and nonlinguistic approaches to MT with respect to several translation aspects: simplicity, accuracy, and multilingualism. GRMT integrates the best features of each approach. The GRMT process is relatively simple and straightforward, not unlike the direct method. However, GRMT is more concerned with preserving linguistic information to produce an accurate translation result, like the transfer approach. GRMT also treats the source language (SL) and target language (TL) separately for easy management in multilingual MT systems, like the interlingual approach.

Unlike the transfer or interlingual approaches, GRMT generates a translation candidate (TC) directly from the input to avoid information loss during the transfer process of the transfer approach and to avoid difficulties in defining neutral concepts for different languages in the interlingual approach (Cercone and Naruedomkul 1997). To ensure that the TC can be generated quickly, simply, and efficiently, GRMT generates the TC by considering the differences between language pairs in terms of syntax and semantics without performing any sophisticated analysis. GRMT analyzes the TC to verify its accuracy. The TC will be repaired, if required, according to the diagnosis which is indicated in the analysis stage. Subsequently, the repaired TC will be analyzed to determine if it still has a sufficiently different meaning from the SL. The analysis and repair processes iterate until the TC conveys the same meaning as the SL. These two stages ensure the accuracy of the translation result. Based on these notions, GRMT (Figure 1) is composed of three phases: Analysis Lite Machine Translation (ALMT), Translation Candidate Evaluation (TCE), and Repair and Iterate (RI).

GRMT takes into account the differences between languages in a unique way, hence a further advantage. If languages can be grouped according to various characteristics—for example, plurality, tenses, passive voice, etc.—which they have in common, then the translation between groups can be performed more simply by GRMT. For example, Group 1 consists of English, French, and Spanish; Group 2 consists of Chinese, Japanese, and Thai. To perform the translation between these two groups, the transfer approach requires six SL

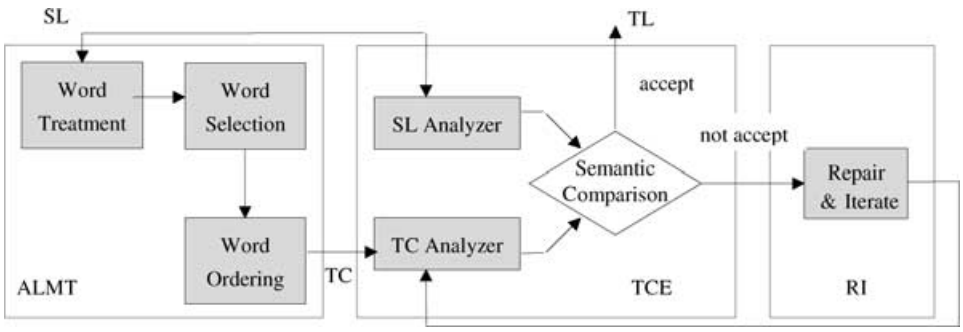


FIGURE 1. GRMT Architecture.

analyzers, six TL generations, and 18 sets of transfer rules, whereas GRMT requires six analyzers and two sets of constraint applications.

An English–Thai translation system has been developed based on the GRMT approach. Our initial experiments of ALMT (Naruedomkul, Cercone, and Sirinaovakul 1999) indicate that the translation candidates can be generated quickly with relatively high accuracy. We have greatly expanded the dictionaries used in the earlier experiment fivefold. All necessary knowledge bases required were updated. ALMT has been improved to increase the performance of GRMT. The English–Thai translation system we developed has been used to translate a number of sentences and phrases. These sample sentences and phrases were collected from different sources: newspapers, textbooks, articles, fairy tales, and fortune cookies. This article presents the design and some experiments of GRMT.

## 2. ANALYSIS LITE MACHINE TRANSLATION

To generate an appropriate translation candidate from the source language, ALMT performs generation in three phases: word treatment, word selection, and word ordering.

*Word Treatment*. Two steps are performed in this phase: source language constraints application and dictionary look-up (Naruedomkul and Cercone 1997). The SL constraints are applied to narrow the scope of possible TL words that correspond to each SL word. Dictionary look-up maps all corresponding words in the TL to each SL word.

SL constraints are characteristics of the SL which differ from those of the TL. Some constraints that are considered in the English–Thai translation system, e.g., plurality,[1] continuous tense, passive voice, and negation, are illustrated in Table 1.

Since Thai word form does not have an inflection resulting from plurality or tense, the inflections of plural noun and present participle forms in English are removed in this step. Once the inflections are removed, the features *plural* and *ing* are added to preserve the meaning of the original sentence. The structure of a passive voice is examined as well; the verb inflection and auxiliary *to be* are removed. The feature *passive* is added to indicate the passive voice. The features *plural*, *ing*, and *passive* will be replaced with appropriate corresponding TL words in the ordering step of ALMT.

[1]This constraint does not apply in the case of the irregular plural forms, e.g., children, feet, men, women, and so on. The irregular plural forms are entries in the dictionaries.

TABLE 1. Some SL Constraints considered in the English–Thai MT system

| SL Constraints | Descriptions | Examples |
|---|---|---|
| Plurality | noun_(e)s → noun + plural | The books . . . → The book + plural |
| Continuous tense | V to be + V_ing → V + ing | I am swimming. → I + ing + swim |
| Passive voice | be + V3 → passive + V | He was arrested. → He passive + arrest |
| Negative | do + not → not | He does not eat → He not eat |

The auxiliary *to do* which precedes a negative *not* is discarded because it is not required in the same expression in Thai. Figure 2 illustrates the outputs of SL constraints applications on Example 1. The inflection $-s$ of the word *symbol* in Example 1 is removed and the feature *plural* is added, the passive voice form *are used* is replaced with *passive use*, the auxiliary *do* is discarded, and the continuous tense form is replaced with *ing talk*.

After the SL constraints are applied, each SL word will be used as a keyword to search for its corresponding words in the TL. If the keyword used is found in the SL-TL dictionary, all possible corresponding TL words will be attached to that SL word. If the keyword is not found, inflectional analysis is performed before searching again. The inflections of each entry are not entries in the dictionaries; therefore the size of the SL dictionary and the search times are reduced. An output of this step is a list of all possible corresponding words of SL words in TL with their WordAsso numbers.

*Word Selection*. The criterion used in the word selection process is the semantic relationship between words. The semantic relationship we developed is a link between word association numbers (WordAsso) (Naruedomkul and Cercone 1999). WordAsso is a number assigned to each word class. WordAsso is our modification to the "A Kind Of" (AKO) information which was designed to be used in "The research and development cooperation project on a machine translation system for Japan and its neighboring countries" (CICC 1995), the topic hierarchy for physical objects (Schubert, Goebel, and Cercone 1979), and Hypernym in WordNet 1.6 (Cognitive Science Laboratory 1997). We then further developed our notion of word classification by classifying words into categories based on the characteristics which they share. We were concerned with consistency as well. The improved WordAsso results in more accurate word selections than the earlier version reported in Naruedomkul and Cercone (1997).[2] In addition, we classify words in any language under the "same umbrella." Words must be classified according to the same criteria, regardless of language; therefore, the GRMT classification can be applied to a multilingual MT system. The details of the word selection process can be found in Naruedomkul and Cercone (1997).

Example 1:
SL: Algebraic symbols are used when you do not know what you are talking about.
SL Cons- output: Algebraic plural symbol passive use when you not know what you ing talk about.

FIGURE 2. SL constraints applications examples.

[2]In that version, Word Association Number is refereed to as the AKO number.

*Word Ordering*. There are two steps in this phase, word addition and word ordering. The syntactic level differences between SL and TL are reconsidered at this point to complete the TC. Some words that are necessary in the target language, not only to retain the meaning of source language, but also to make them grammatically correct, are added into the string before the ordering can be performed. In the case of noninflection languages, e.g., Thai, Chinese, Japanese, etc., the *past tense* and the *plurality* are expressed by additional words. In the case of plurality, a *classifier* is also required. Classifiers are used to express not only a quantity but also to modify a noun (Sornlertlamvanich, Pantachat, and Meknavin 1994). Each noun relates to a specific classifier; therefore, the classifier relation was designed in the form of WordAsso to be used in selecting the appropriate classifier for each noun. Details and examples can be found in Naruedomkul and Cercone (1999).

After all necessary words have been added, ALMT rearranges the selected words in grammatical order according to the ordering rule without performing any analysis. The structure of the Thai phrase is similar to that of the English phrase in that the typical sentence contains subject, verb, and object, in that order. However, some structures are different, e.g., in Thai the head (noun) must precede its attributes, possessive pronouns, and determiners. The negative of *can*, *could*, and *must* is formed in the reverse order of the order in English.

Table 2 presents the results of applying ALMT to Example 2. The SL of Example 2 is shown in the first column. None of constraints are applied in Example 2 as illustrated in the second column. Each word in the second column is used as a keyword to search for the corresponding words in Thai. All possible meanings of each SL word are shown in the third column. Some of them have more than one meaning, e.g., *old*, *in*, *live*, *with*, and so on. The appropriate meaning of *old* and *in* can be selected by considering the semantic relationship between words and the choice for each of these is shown in the fourth column. However, appropriate words for *live* and *with* cannot be selected in the same manner because there is no explicit relationship between these words and words in their proximity (according to the WordAsso relationship template). Therefore, the first meaning appearing in the list of meanings for each word is selected. All selected words are shown in the fourth column. The word ได้ (dâj), the fifth column, is added to clarify the past tense (lived). The classifiers คน (khon), หลัง (laù$\eta$), and ตัว (tua) are also added according to Thai grammar. The indefinite determiners *a* and *an* in this expression correspond to the word หนึ่ง (ny`$\eta$) in Thai indicate the need for classifiers for the words ผู้หญิง (phûjiù$\eta$-woman), แมว (m$\varepsilon\varepsilon$w-cat) and ไก่ (kàj-hen), respectively. The word ผู้หญิง (phûjiù$\eta$-woman) belongs to the class Female (1-1-1-1-1-2), a subclass of Human (1-1-1-1). A noun that belongs to the class 1-1-1-1 is compatible with a classifier with the WordAsso

Example 2: An old woman lived in the cottage, with a fat black cat and a plump brown hen.

TC: ผู้หญิง แก่ คนหนึ่ง ได้ อยู่ ใน กระท่อม หลัง นั้น กับ แมว สีดำ อ้วน ตัว หนึ่ง และ ไก่ สีน้ำตาล อวบ ตัว หนึ่ง[3]
CT: ผู้หญิง แก่ คนหนึ่ง ได้ อยู่ ใน กระท่อม หลัง นั้น กับ แมว สีดำ อ้วน ตัว หนึ่ง และ ไก่ สีน้ำตาล อวบ ตัว หนึ่ง
(phûuji$\check{}\eta$[4]-woman) (k$\varepsilon$`$\varepsilon$-old) (khon-clas) (ny`$\eta$-an) (dâj-past) (ju`u-live) (naj-in) (krath$\partial$`m -cottage) (la$\check{}\eta$-clas) (nán-the) (kàb-with) (m$\varepsilon\varepsilon$w-cat) (si$\check{}$idam-black) (?ûan-fat) (tua-clas) (ny`$\eta$-a) (l$\varepsilon$`-and) (kàj-hen) (si$\check{}$inámtaan-brown) (?ùab-plump) (tua-clas) (ny`$\eta$-a)

[3]Thai language is written in a string of words with no explicit boundary marker;therefore, the different segmentations result in the different meanings or no meaning. In this article, awhite space is used to specify the word boundary to make it clear to the reader.

[4]Phonetic transcription of Thai provided in Thai–EnglishStudent's Dictionary compiled by Mary R. Haas (Hass 1964).

TABLE 2. ALMT Steps Applied to the Sentence of Example 2.

| English | Constraint Application Output | Word Treatment Output | Selected Word in Thai | Word Addition | Word Ordering |
|---|---|---|---|---|---|
| An | An | หนึ่ง (ny`ŋ) | หนึ่ง (ny`ŋ) | | ผู้หญิง (phûujiˇŋ) |
| | | | | คน (khon) | แก่ (kɛ`ɛ) |
| old | old | แก่ (kɛ`ɛ), เก่า (kàw) | แก่ (kɛ`ɛ), | | คน (khon) |
| woman | woman | ผู้หญิง (phûujiˇŋ) | ผู้หญิง (phûujiˇŋ) | | หนึ่ง (ny`ŋ) |
| | | | | ได้ (dâj) | ได้ (dâj) |
| lived | live | อยู่ (ju`u), มีชีวิต (miichiiwíd), ดำรงชีวิต (damroŋchiiwíd) | อยู่ (ju`u) | | อยู่ (ju`u) |
| in | in | ใน (naj), เข้ามา(khâwmaa), เข้าไป(khâwpaj), อยาง(jàaŋ) | ใน (naj) | | ใน (naj) |
| the | the | นั้น (nán) | นั้น (nán) | | กระท่อม (krathɔ̂m) |
| | | | | หลัง (laˇŋ) | หลัง (laˇŋ) |
| cottage | cottage | กระท่อม (krathɔ̂m) | กระท่อม (krathɔ̂m) | | นั้น (nán) |
| with | with | กับ (kàb), ด้วย (dûaj), ซึ่งมี (syˆŋ) | กับ (kàb) | | กับ (kàb) |
| a | a | หนึ่ง (ny`ŋ) | หนึ่ง (ny`ŋ) | | แมว (mɛɛw) |
| | | | | ตัว(tua) | สีดำ (siˇidam) |
| fat | fat | อ้วน (?ûan), ไขต้น (khaˇjman) | อ้วน (?ûan) | | อ้วน (?ûan) |
| black | black | ดำ (dam), สีดำ (siˇidam) | สีดำ (siˇidam) | | ตัว (tua) |
| cat | cat | แมว (mɛɛw), ผู้หญิวนม่ม้ (phûujiˇŋmâjdìi) | แมว (mɛɛw) | | หนึ่ง (ny`ŋ) |
| and | and | และ (lɛ`) | และ (lɛ`) | | และ (lɛ`) |
| a | a | หนึ่ว (ny`ŋ) | หนึ่ว (ny`ŋ) | | ไก่ (kàj) |
| | | | | ตัว (tua) | สีน้ำตาล (siˇinámtaan) |
| plumb | plumb | อวบ (?ùab) | อวบ (?ùab) | | อวบ (?ùab) |
| brown | brown | สีน้ำ ตาล(siˇinámtaan), เกรียม(kriam), คดำ (khlám) | สีน้ำ ตาล(siˇinámtaan) | | ตัว (tua) |
| hen | hen | ไก่ (kàj) | ไก่ (kàj) | | หนึ่ง (ny`ŋ) |

clf_rel('**1-1-1-1**',[[wasso('คน',['**2-4-2-1-1-1**'])]]).
clf_rel('**1-1-1-2**',[[wasso('ตัว',['**2-4-2-1-1-2**'])]]).
clf_rel('1-1-2-1-6-1',[[wasso('แห่ง',['2-4-2-1-2-2'])]]).
clf_rel('2-4-3-1-2',[[wasso('วัน',['2-4-2-3'])]]).
clf_rel('**1-1-2-1-1-2-2**',[[wasso('หลัง',['**2-4-2-1-2-10**'])]]).

FIGURE 3. Examples of classifier relations.

number 2-4-2-1-1-1 based on the classifier relation illustrated in Figure 3. Therefore, the classifier คน (khon) with 2-4-2-1-1-1 is selected for the word ผู้หญิง (phûujiù$\eta$-woman). The words แมว (m$\varepsilon\varepsilon$w-cat) and ไก่ (kàj-hen) belong, respectively, to the classes mammal (1-1-1-2-1-1) and fowl (1-1-1-2-1-2-2). Both are subclasses of animal (1-1-1-2). Because a noun with WordAsso number 1-1-1-2 relates to a classifier with 2-4-2-1-1-2 according to the classifier relation shown in Figure 3, the classifier ตัว (tua) with 2-4-2-1-1-2 is selected for the words แมว (m$\varepsilon\varepsilon$w-cat) and ไก่ (kàj-hen).

The definite determiner *the* corresponds to the word นั้น (nán) in Thai and indicates the need for classifiers for the word กระท่อม (krath$\partial$ˆm-cottage). The word กระท่อม (krath$\partial$ˆm-cottage) belongs to the class Housing (1-1-2-1-1-2-2) which is compatible with a classifier with the WordAsso number 2-4-2-1-2-10 based on the classifier relation illustrated in Figure 3. Therefore, the classifier หลัง (laˇ$\eta$) with the WordAsso 2-4-2-1-2-10 is selected for the word กระท่อม (krath$\partial$ˆm-cottage).

The selected words in each noun phrase, *an old woman*, *the cottage*, *a fat black cat*, and *a plump brown hen* are rearranged into Thai grammatical order as illustrated in the last column of Figure 3. The generated TC for Example 2 is exactly the same as the *Correct Translation* (CT).

## 3. TRANSLATION CANDIDATE EVALUATION

The second phase of GRMT, TCE, analyzes the generated translation candidate to determine whether the TC retains the meaning of the source language. TCE analyzes both the SL and the TC in parallel, then compares the parses semantically alone, because there are syntactic level differences between languages. If the semantic results are the same, the TC will be deemed an appropriate translation. If the semantic results are different, the TC will be repaired in the third phase, Repair and Iterate. TCE comprises two modules: the analyzer and semantic comparison.

*The Analyzer*. The analysis module analyzes the TC to examine its syntax and semantics: whether the TC is grammatically correct according to the TL grammar and whether the TC retains the meaning of the SL. Therefore, two steps are performed by the analyzer, *Parsing* and *Semantic Extraction*.

*Parsing* is applied to both the SL and the TC. In our implementation, we have developed grammars for English and Thai based on Head-Driven Phrase Structure Grammar (HPSG) (Pollard and Sag 1994). Our grammar is a modification of the grammars that were provided by Gerald Penn (Penn 1994) and Colin Matheson (Matheson 1996). The grammars we developed have been implemented using the Attribute Logic Engine (ALE) version 3.2 Beta. ALE is an integrated phrase structure parsing and definite clause logic programming system in which the

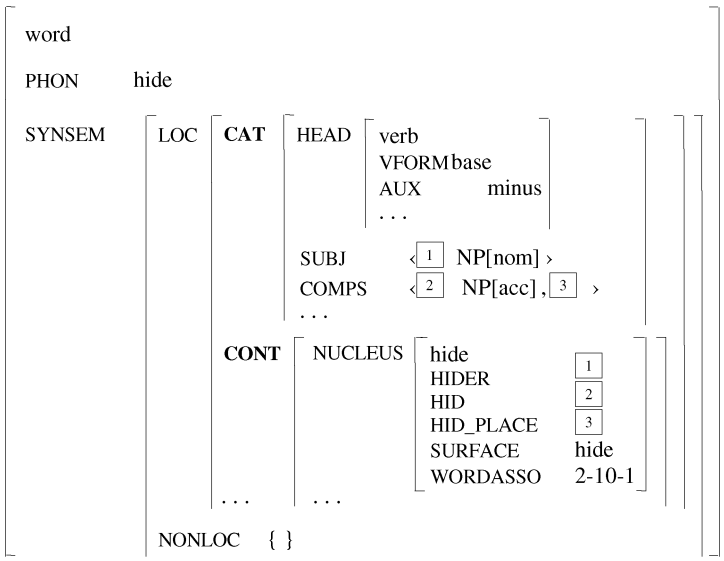


FIGURE 4. The lexical entry for *hide.*

terms are typed as feature Structures (Carpenter and Penn 1999). Key advantages of HPSG are the rich information lexicon, small number of grammar rules, and the structure sharing. HPSG views the grammatical categories as complexes with internal structure which allow us to say that the two categories are the same in certain aspects, while remaining different in others. Therefore, HPSG gives each word a category built up from a set of features called the Feature Structure. Feature Structure is a description of an object; it specifies some or all of the information that is asserted to be true of the object. The features CATEGORY (CAT) and CONTENT (CONT) represent the syntactic and semantic information of an object, respectively. Figure 4 represents a partial description of the lexical entry *hide* in an attribute-value matrix (AVM) diagram.[5] This representation indicates that *hide* is a word.[6] Its syntactic category is a verb in uninflected form (verb, VFORM:base). It is not an auxiliary verb (AUX:minus). A subject of *hide* is a (nominal) noun phrase[7] (SUBJ:$NP_{[nom]}$) and this subject must unify with the CONT value of *hide*, HIDER (indicated by index 1). The first complement of *hide* is an accusative noun (COMPS:$NP_{[acc]}$) and it must unify with the CONT value of *hide*, HID. The second complement must unify with the CONT value of *hide*, HID_PLACE. Figure 5 illustrates the parse of the SL in Example 3 and the parse of its TC. The TC is generated by ALMT. The parses are in ALE representation. Both SL and TC are licensed by our grammars. Their syntax are shown in the dotted box. Further details on grammar rules applications can be found in Naruedomkul (2000).

*Semantic Extraction* extracts the semantic information of the SL and the TC from their parses so that we can cross-examine the meaning of the TC with that of the SL. The semantic

[5]AVM is the standard method of representing grammatical information in modern computation grammar theories.

[6]Utterances in HPSG are modeled in terms of feature structures of type *sign*, with its two immediate subtypes *word* and *phrase* (Ginzburg and Sag 1998).

[7]For English, nouns are classified into nominative (nom) and accusative (acc).

Example 3: The ugly duckling hides his head under his wing.

SL: The ugly duckling hides his head under his wing.

TC: ลูกเป็ด ขี้เหร่ ตัว นั้น ซ่อน หัว ของเขา ใต้ ปีก ของเขา
(lûugpèd-duckling) (khîirèe-ugly) (tua-clas) (nán-the) (sɔ̂ɔn-hide) (hŭa-head)
(khɔ̌ɔŋkhaw-his) (tâaj-under) (pìig-wing) (khɔ̌ɔŋkhaw-his)

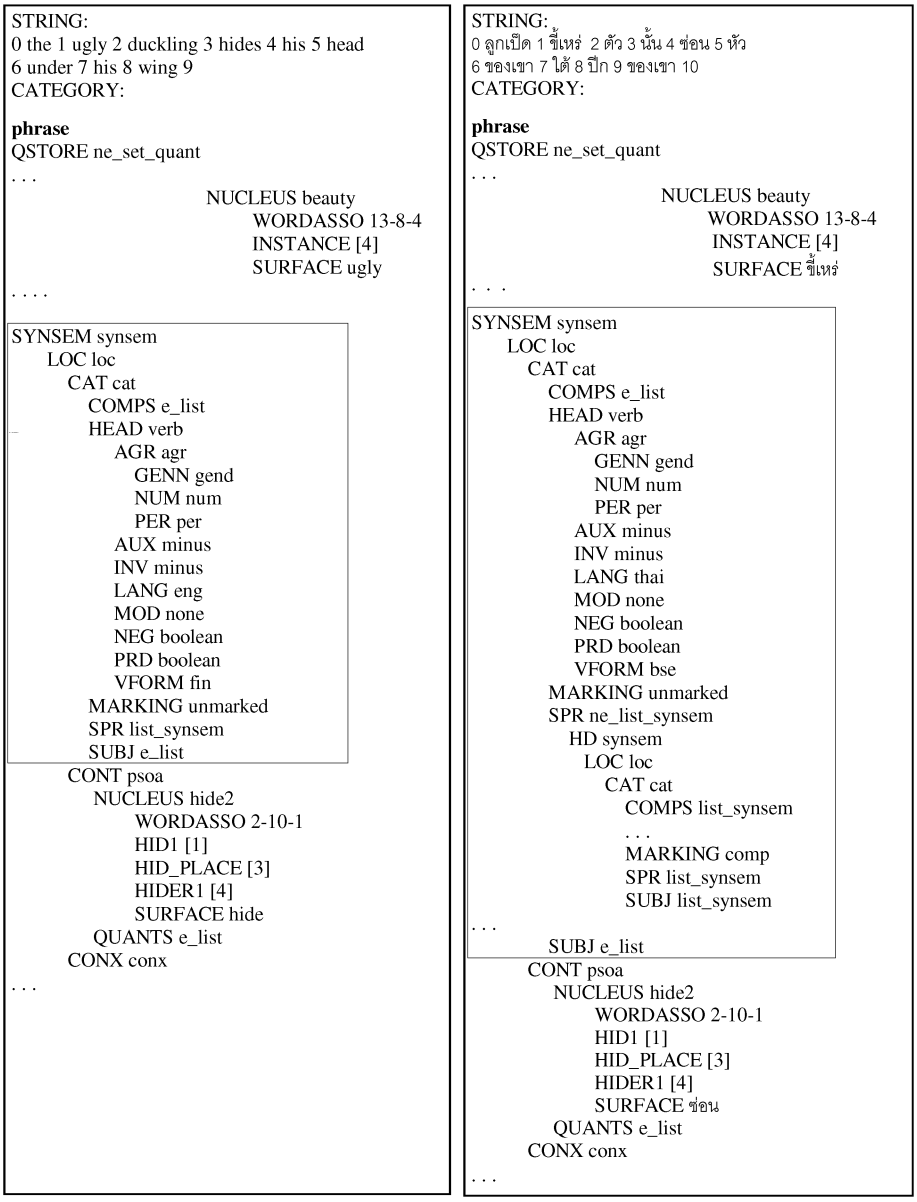


FIGURE 5. The syntax of Example 3 and that of its translation.

representation of an expression in GRMT is generally based on the representation provided by Pollard and Sag (1994) and Sag and Wasow (1999). The features CONT, CONX and QSTORE hold the semantics of the object. The semantic representation described in CONT value of *hide* (Figure 4) corresponds to the situation in which□1 (*the hider*) hides□2 (*the hid)* in (or under, from, etc.)□3 (*the hid_place*). The features SURFACE and WORDASSO indicate the word form and WordAsso number of the lexical entry. The features SURFACE and WORDASSO are used in the Repair and Iterate phase. The feature QSTORE is a storage for the quantifiers. The feature CONX contains linguistic information that bears on certain context-dependent aspects of semantic interpretation. Examples of QSTORE and CONX values can be found in Figure 11. Further details can be found in Pollard and Sag (1994).

*Semantic Comparison*. In comparing the semantics between the SL and its TC, the values of the features CONT, QSTORE, and CONX are considered. If the values of these features of the parsed SL are the same as those of the parsed TC, TCE concludes that the TC does not require repair. If any of these features are different, TCE will provide the information of the parsed SL which differs from that of the TC parse. This information will be used in the next phase, Repair and Iterate. The comparison process begins by investigating the CONT value of both SL and TC parses. Figures 6 and 7 illustrate the simplification representation of semantic information of the SL and that of the TC in Example 3. The CONT values of both parses represent the same kind of relation involved, *hide2*, and the same types of persons or things who/which are participating in this relation, HIDER, HID, and HID_PLACE. Once the CONT values of the SL parse and that of the TL parse are recognized as the same, TCE then cross-examines the QSTORE and the CONX values of the SL parse with those of the TC parse to determine whether each variable (indices) in the CONT value is associated to the same object. The index□1 in the CONT value of both SL and TC parses is associated with a *bodypart* in the class of 1-4-1-1-1 (in the QSTORE value), it is *singular* and referred to as *head* in the SL and as หัว (hŭa) in the TL. หัว (hŭa) is the translation of the word *head* in Thai. The index□2 corresponds to a *bodypart* in the class of 1-4-1-1-3-1, it is singular and referred to as *wing* in the SL and ปีก (pìig) in the TL. Again, ปีก (pìig) is a translation of *wing* in Thai. The last index□3 is associated with an animal in the class of 1-1-1-2-1-2-1 which is *singular* and referred to as *duckling* in the SL and ลูกเป็ด (lûugpèd) in the TL. ลูกเป็ด (lûugpèd) is the translation of the word *duckling* in Thai. The CONX feature in this example contains no value. According to the comparison process, TCE found no difference between the CONT, QSTORE, and CONX of the SL parse and those of the TC parse. Therefore, the TC is deemed as an appropriate translation for the SL in the Example 3.

The CONT values of the SL parse and the TC parse in Example 4 (Figure 8) represent the same relation involved, *cover2*, and the same types of persons or things who/which are participating in this relation, COVERED, COVERED_PLACE and COVERER1. The SURFACE value in the CONT of the SL parse is *cover* and that of the TC is คลุม (khlum). The value of the feature SURFACE is a word form. คลุม (khlum) is a translation of *cover*. However, their WORDASSO values are different, *cover* belongs to the class of 2-1-19 whereas คลุม (khlum) belongs to the class of 2-1-19-2 as illustrated in Table 3. Even though *cover* and its translation คลุม (khlum) carry the same meaning, they are different in the language usage. Differences in languages and cultures result in different circumstances of language usage. The word *cover* in *John covers his ears with his hands* corresponds to ปิด (pìd) in Thai. Therefore, when verifying the meaning of *cover*, the word and its translation are considered no different if they belong to the same superclass.

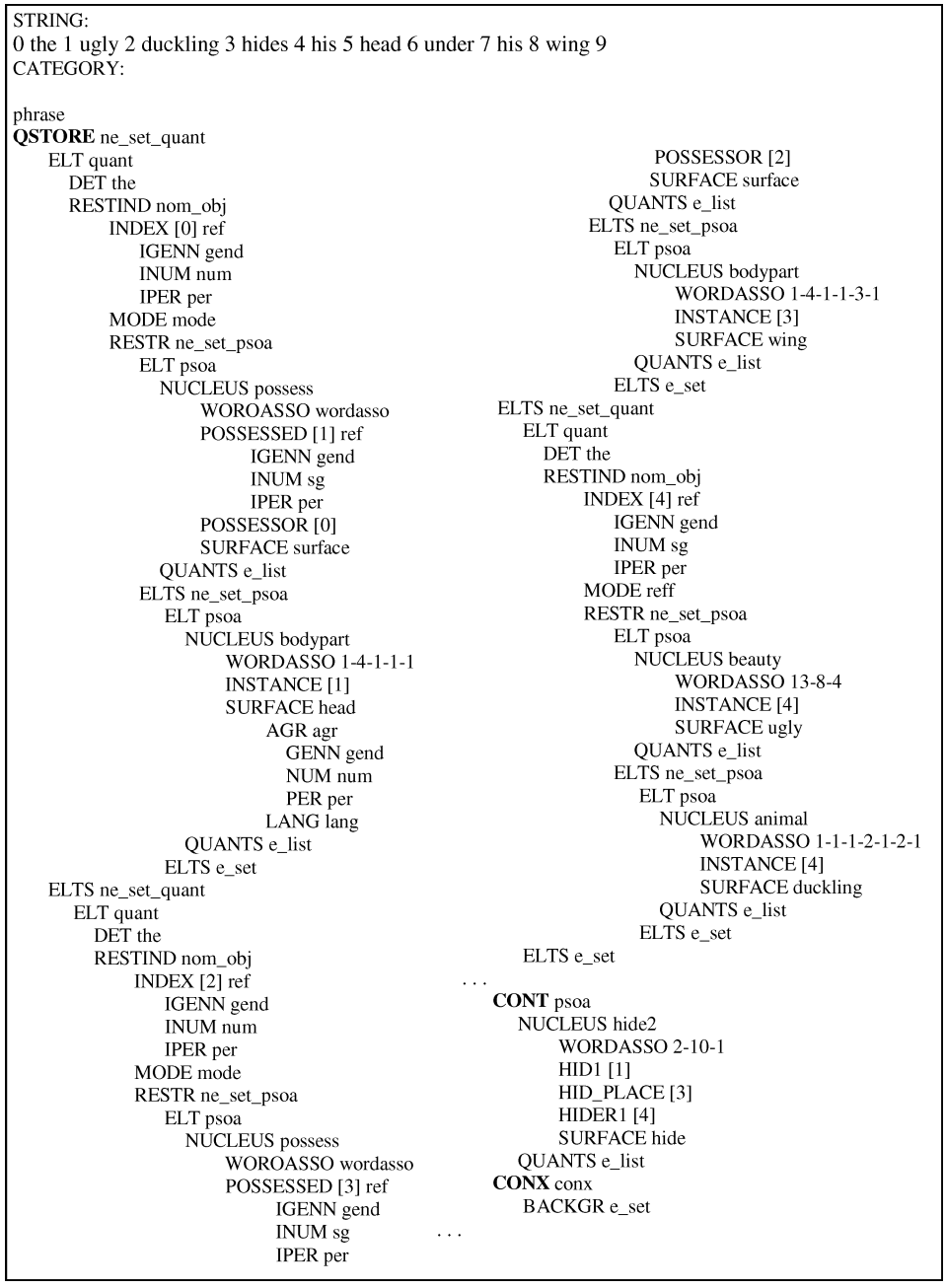


FIGURE 6. The semantic information of the SL parse in Example 3.

## 4. REPAIR AND ITERATION

The idea of the RI phase is to perform the repair process if it is required and then return the repaired translation candidate to the TCE phase. TCE reanalyzes the repaired TC to determine if a different meaning from the source language remains.

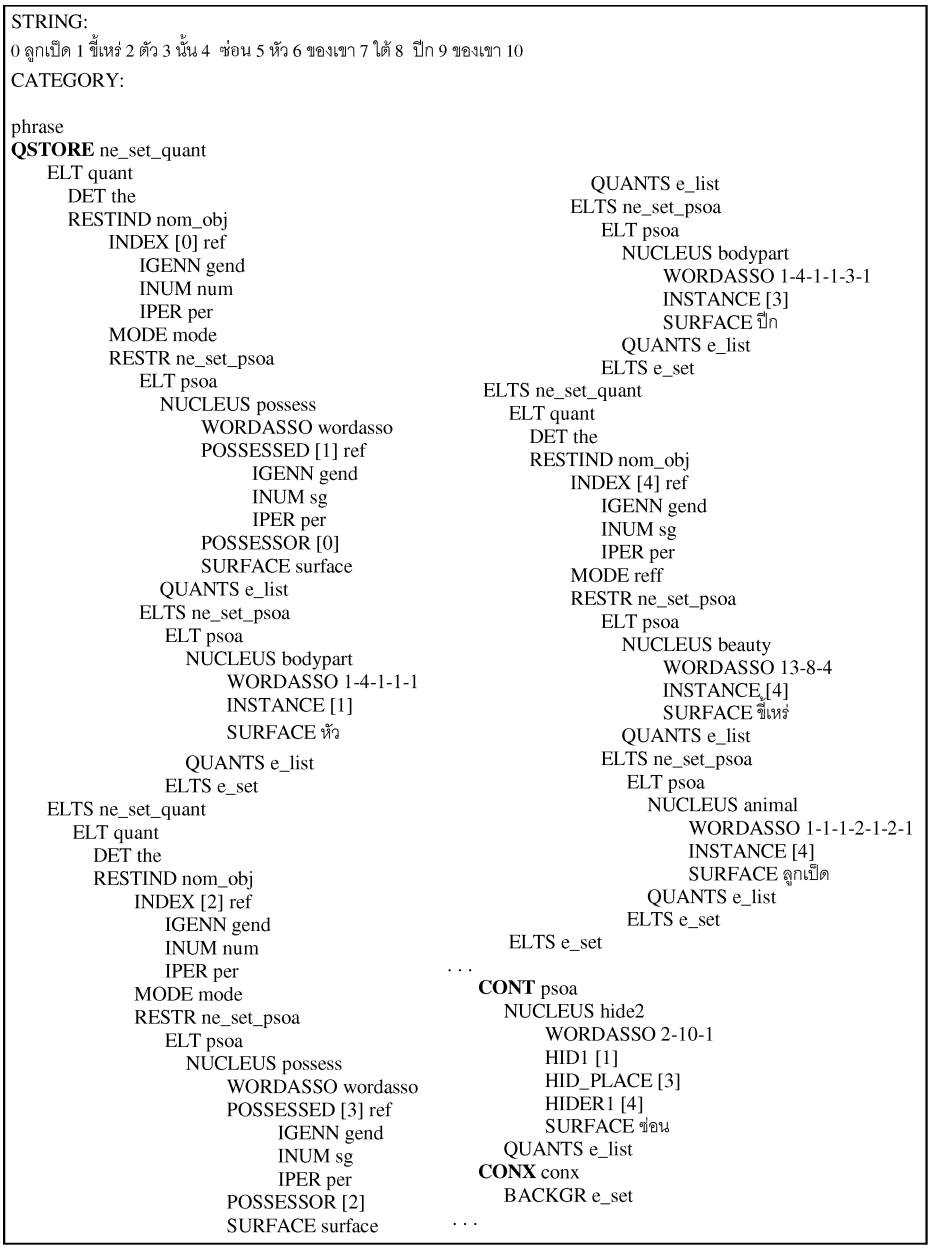


FIGURE 7. The semantic information of the TC parse in Example 3.

RI examines the result of TCE. The TCE output is the TC with the semantic information of the SL which differs from that of the TC. With this information, RI is able to detect the part of the TC which causes the mistranslation. The mistranslated part will be replaced with a more appropriate translation. RI searches the *Word Treatment output* for a more appropriate translation based on the information provided by TCE. The TC with the new selection is then

Example 4:

SL: John covers his head with his coat.
TC: จอห์น คลุม หัว ของเขา ด้วย เสื้อคลุม ของเขา
(cɔɔn-John) (khlum-cover) (huˇa-head) (khɔˇɔŋkhaw-his) (dûaj-with)
(syˆakhlum-coat)( khɔˇɔŋkhaw-his)

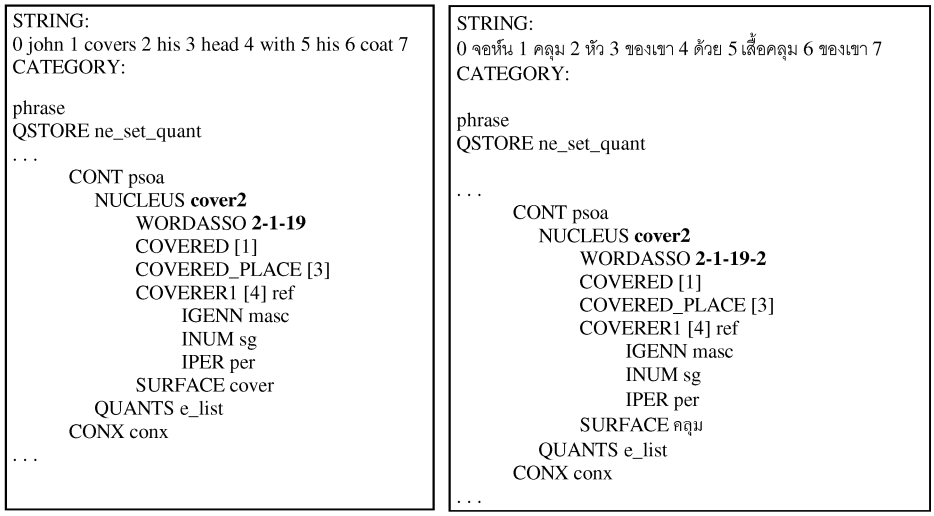


FIGURE 8. The CONT value of the SL parse and that of the TC of Example 4.

put through the *Word Ordering* module to revise its syntax. Once the revision is completed, the repaired TC is returned to TCE.

In the case that the CONT or the QSTORE value of the SL is different from that of the TC, the SL CONT value: SURFACE and WORDASSO features are passed to RI. The SURFACE value indicates the surface form of the word which causes the mistranslation. The WORDASSO value specifies the proper meaning of the word in question in terms of WordAsso number. Therefore, to repair the CONT or QSTORE value of the TC, RI reselects the corresponding word in the TL for the word specified in the SURFACE value. The reselection is done by searching the *Word Treatment Output* for the corresponding word which has the same WordAsso number as specified in the WORDASSO value.

In the case that the CONX value of the SL differs from that of the TC, the SL CONX value: the BEARER and NAME features are passed to RI. The BEARER value specifies the

TABLE 3. Class of *cover*.

| WordAsso and Class description | Word |
|---|---|
| 2-1-19 Cover as to place something on or over to protect or hide | cover |
| 2-1-19-1 Cover as to cover eyes, ears | ปิด (pìd) |
| 2-1-19-2 Envelop | ดคลุม (khlum) |

Example 5:

SL: The big ugly duckling likes Joan.
TC: ลูกเป็ด ขี้เหร่ โต ตัว นั้น เหมือน โจแอน
(lûugpèd-duckling) (khîilèe-ugly) (too-big) (tua-clas) (nán-the) (myˇan-like) (joo?ɛn-Joan)

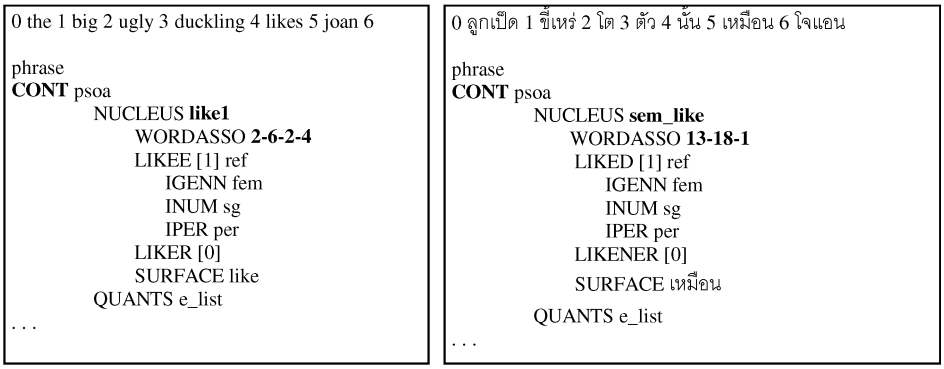


FIGURE 9. The CONT values of the SL and that of the TL parses.

index which associates with the certain name specified in the NAME value. RI repairs the TC by associating the right names to the right indices based on the information provided by TCE.

The CONT values of the SL and the TC parses of Example 5 are different as illustrated in Figure 9. The CONT value of the SL indicates that the word *like* is mistranslated (SURFACE: like) and its proper meaning in this expression is *to regard with pleasure or fondness* which is classified into the class of 2-6-2-4 (WORDASSO: 2-6-2-4). Therefore, RI begins the repair process by re-selecting the translation of the word *like.* RI searches the *WordTtreatment output* (Table 4) for the translation of *like* which has WordAsso number 2-6-2-4 and thus, the translation ชอบ (chɔ̂ɔ), is selected.

The word เหมือน (myˇan), in the TC is then replaced with ชอบ (chɔ̂ɔ). The TC with the new selection (Figure 10) is put through the *Word Ordering* module to revise its syntax. Once the repair processes successfully, the repaired TC is analyzed by TCE. Figure 11 illustrates the semantic information of the SL parse and that of the repaired TC. Their CONT, QSTORE, and CONX values are the same. The repaired TC is then deemed as an appropriate translation

TABLE 4. A Part of Word Treatment Output.

| English | Word Treatment Output | Description |
|---|---|---|
| like | ชอบ (chɔ̂ɔ), 2-6-2-4 | to regard with pleasure or fondness |
| | เหมือน (myˇan), 13-18-1 | in the same way as; with the same quality as |
| | คล้าย (khláaj), 13-18-2 | similar to; almost but not exactly the same |

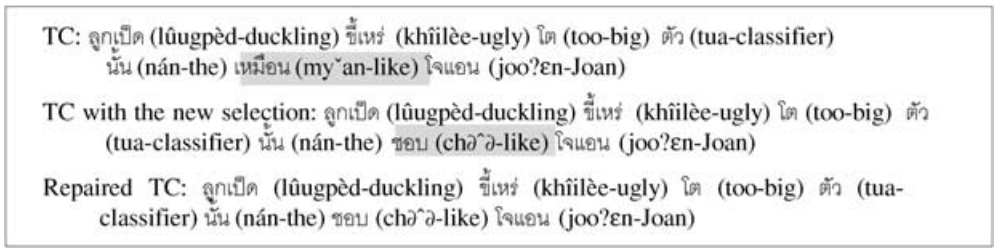


FIGURE 10. The repaired TC of Example 5.

for the SL in Example 5. Further details of the repair process can be found in Naruedomkul (2000).

## 5. CONCLUDING REMARKS

GRMT consists of three phases: ALMT, TCE, and RI. ALMT generates the translation candidate in a simple, straightforward manner, similar to the direct approach, but ALMT is more efficient because it accounts for differences between language pairs in terms of both syntax and semantics and this analysis ensures that the generated TC is exact or close to the correct translation.

The TCE analyzes the TC to determine whether its syntax and semantics are grammatically correct and whether it conveys the meaning of the original sentence. If the TC does not, RI will repair it. These two stages, TCE and RI, ensure accuracy of the translation.

GRMT treats the SL and TL separately, as is also the case with the interlingual approach. GRMT is also aware of the differences between languages. Therefore, if languages can be grouped according to various characteristics, for example, plurality, continuous tenses, passive voice, etc., which they have in common, then the translation between groups can be performed more simply by GRMT.

Another aspect of concern in designing an MT system is the structure of the knowledge base, e.g., the constraints and the WordAsso information in the dictionary. The structure of each knowledge base component should be direct, intuitive, and easy to extend to a large-scale MT system. In generating a reliable TC, ALMT requires simple information. This simplicity ensures that knowledge bases required in the ALMT phase are easy to manage in a large-scale MT effort. The dictionary used in earlier reported experiments of ALMT (Naruedomkul and Cercone 1997) was expanded to 348 English words and 731 Thai words. The "semantic relationship" and the "ordering rule" were also updated. There has been no appreciable increase in processing time when running ALMT using this larger dictionary. The lexicons used in the TCE for both SL and TL are informative and easy to manage.

## ACKNOWLEDGMENTS

The authors are members of the Institute for Robotics and Intelligent Systems (IRIS) and wish to acknowledge the support of the Networks of Centres of Excellence Program of the Government of Canada, the Natural Sciences and Engineering Research Council (NSERC), and the participation of PRECARN Associates, Inc.

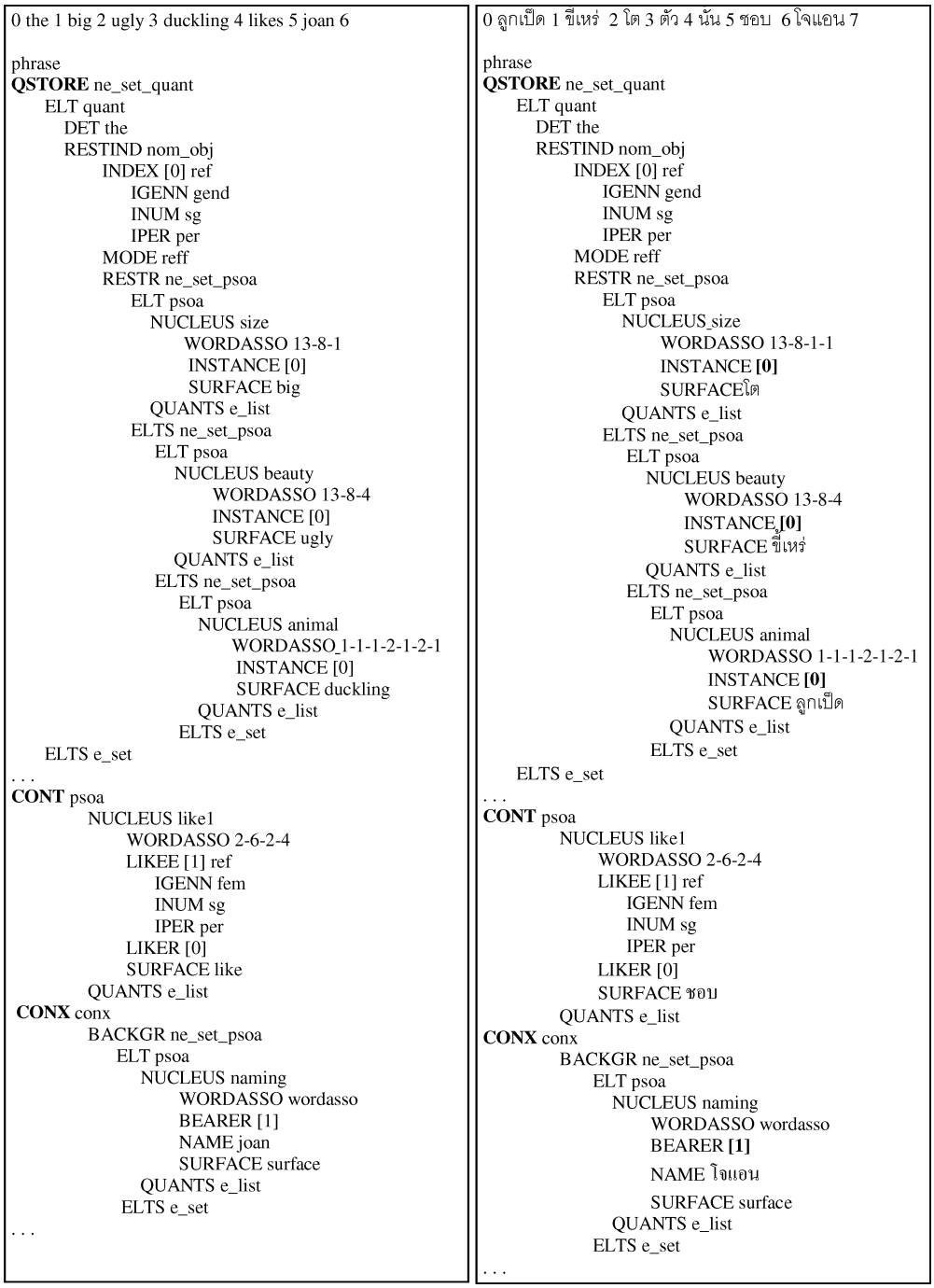


FIGURE 11. The semantic information of the SL and that of the repaired TC.

## REFERENCES

CARPENTER, B., and G. PENN. 1999. ALE: The Attribute Logic Engine User's Guide Version 3.2 Beta. http://www.sfs.nphil.uni-tuebingen.de/~gpenn/ale.html#Obtain, accessed in May 2002.

CERCONE, N., and K. NARUEDOMKUL. 1997. Why GRMT? Vivek, **10**(3):12–15.

CICC. 1995. Thai Analysis Rules. Technical Report 6-CICC-MT46, Machine Translation System Laboratory, Center of the International Cooperation for Computerization, Tokyo.

COGNITIVE SCIENCE LABORATORY, PRINCETON UNIVERSITY. 1997. WordNet-A Lexical Database for English. http://www.cogsci.princeton.edu/~wn/, accessed in December 2002.

GINZBURG, J., and I. A. SAG. 1998. English Interrogative Constructures (Draft of July) *In* Construction: An HPSG Perspektive. *Edited by* I. A. SAG and A. KATHOL. Language Advanced Course, 10th European Summer School in Logic, Language and Information, Saarbrücken, 17–28 August.

HAAS, M. R. 1964. Thai–English Student's Dictionary. Stanford University Press, Stanford, CA.

MATHESON, C. 1996. HPSG in ALE. http://www.ltg.hcrc.ed.ac.uk/projects/ledtools/ale-hpsg/, accessed in April 2002.

NARUEDOMKUL, K. 2000. Beyond Intralingual: Squeezing More Accuracy into Machine Translation. PhD thesis (Draft), Department of Computer Science, University of Regina, Canada.

NARUEDOMKUL, K., and N. CERCONE. 1997. Steps Toward Accurate Machine Translation. *In* Proceedings of the 7th International Conference on Theoretical and Methodological Issues in Machine Translation, Santa Fe, NM, pp. 63–75.

NARUEDOMKUL, K., and N. CERCONE. 1999. The Role for Word Association Numbers in Machine Translation. *In* Proceedings of the Conference Pacific Association for Computational Linguistics (PACLING'99), Waterloo, Ontario, Canada, pp. 379–392.

NARUEDOMKUL, K., N. CERCONE, and B. SIRINAOVAKUL. 1999. English–Thai Translation: Initial Experiments with a Multiphase Translation System. Computational Intelligence, **15**(2):128–151.

PENN, G. 1994. hpsg.pl. http://www.sfs.nphil.uni-tuebingen.de/~gpenn/ale.html#AleGrammars, accessed in December 2002.

POLLARD, C., and I. A. SAG. 1994. Head-Driven Phrase Structure Grammar. Center for the Study of Language and Information, Stanford. The University of Chicago Press, Chicago, London.

SAG, I. A., and T. WASOW. 1999. Syntactic Theory: A Formal Introduction, CSLI Lecture Notes Number 92. Center for the Study of Language and Information, Stanford, CA.

SCHUBERT, L. K., R. G. GOEBEL, and N. CERCONE. 1979. The Structure and Organization of a Semantic Net for Comprehension and inference. *In* Associative Networks: Representation and Use of Knowledge by Computers. *Edited by* N. V. Findler. Academic Press, NY, pp. 121–175.

SORNLERTLAMVANICH, V., W. PANTACHAT, and S. MEKNAVIN. 1994. Classifier Assignment by Corpus-Based Approach. The Fifteenth Computational Linguistics Symposium (COLING-94), Kyoto, Japan, vol. 1, pp. 556–561.