PRE-PROCESSING TASKS FOR RULE-BASED ENGLISH-KOREAN MACHINE TRANSLATION SYSTEM

Sung-Dong Kim
Dept. of Computer Engineering, Hansung University, 389 Samseon-dong, Seongbuk-gu, Seoul, Republic of Korea

Keywords: Rule-based machine translation, Natural language processing, Pre-processing.

Abstract: This paper presents necessary pre-processing tasks for practical English-Korean machine translation. The pre-processing task consists of a problem that requires pre-processing and a solution for the problem. There are many differences between English and Korean, so it is difficult to resolve the differences using parsing and transfer rules. Also, source sentences often include non-word elements, such as parentheses, quotation marks, and list markers. In order to resolve the differences efficiently and make source sentences appropriate to translation system by arranging them, we propose pre-processing for source sentences. This paper studies various pre-processing tasks and classifies into several groups according to the time when the tasks are performed in English-Korean machine translation system. In experiment, we show the usefulness of the defined pre-processing tasks for generating better translation results.

1 INTRODUCTION

Recent English-Korean machine translation systems generate good translation for relatively short sentences. But there are problems that a practical English-Korean machine translation system must solve. It is difficult to translate long sentences and sentences with special patterns. In rule-based translation, context-free grammar is generally used to represent English syntactic structures. The grammar has limitation to express structures for long sentences consisting of comma-separated sub-sentences and for sentences with special patterns. Especially, the syntactic analysis of sentences with commas is very difficult. It is difficult to try to cover those sentences using syntactic rules. Also, there are many differences between English and Korean, so it is difficult to resolve the differences using parsing and transfer rules. An idiom-based translation approach (Yoon, 1993) is adopted to overcome the differences, where fixed format idioms and phrasal idioms are effective in generating readable and meaningful translation results. Further, they try to translate sentences with special patterns using extended idioms (Kim and Kim, 1998). But the idiom translation approach may cause the side effects in idiom recognition that interfere parsing and result in wrong translations. In practical English-Korean translation, source sentences often include non-word elements, such as parentheses, quotation marks, list markers, and etc. These non-word elements make the syntactic analysis difficult, so they are processed properly before the normal translation process.

This paper studies a *pre-processing* as a method of solving the above problems in rule-based English-Korean machine translation. The target of pre-processing in this paper is an input source sentence from plain documents, rather than formatted documents like HTML ones. The system has rules for lexical analysis, parsing and transfer. It adopts idiom-translation approach to resolve differences between two languages and uses partial parsing method by segmenting source sentences to efficiently translate long sentences. In this paper, we search problems that require pre-processing during the analysis steps: lexical analysis, sentence segmentation, parsing, and transfer. Also, we present the solutions for the problems. A *pre-processing task* consists of a problem and a solution. We classify the pre-processing tasks into groups according to the time when the tasks are performed.

Section 2 briefly surveys other works for pre-processing in English-Korean machine translation. Section 3 presents the pre-processing tasks and their classification. Section 4 shows how many sentences will be benefited by the defined pre-processing tasks. Section 5 concludes the paper with further works.

Kim S..
PRE-PROCESSING TASKS FOR RULE-BASED ENGLISH-KOREAN MACHINE TRANSLATION SYSTEM .
DOI: 10.5220/0003151702570262
In *Proceedings of the 3rd International Conference on Agents and Artificial Intelligence* (ICAART-2011), pages 257-262
ISBN: 978-989-8425-40-9

2 RELATED WORKS

English-Korean machine translation (EKMT) treats two languages that are very different in nature. The differences must be solved for accurate translation. In building an EKMT system, pre-processing is useful in solving the differences and will be applied in various positions through the translation steps. English has hyphenated words. (Yuh *et al.*, 1997) proposed translation method for hyphenated words which uses morphological analysis and considers part-of-speech sequence. In (Yuh *et al.* 1996), they defined the functions of a pre-processor for EKMT. They presented sentence splitting in a given document, words identification (hyphenated words, pronoun, abbreviations, and special symbols), normalizing upper/lower case letters and recognition of composition words (multi-word numeric expression, geographic names, organization names) as major functions of the pre-processor. The above studies were for word-level pre-processing problems and the pre-processor must be positioned before normal translation process.

In translation of long English sentences, sentence segmentation and partial parsing were used (Kim *et al.*, 2001). Also, (Kim, 2008) presented comma rewriting for accurate analysis of long sentences consisting of comma-separated sub-sentences. These are pre-processing of source sentences for efficient and accurate translation of long sentences. They were for phrase/sentence-level problems.

This paper considers above studies, searches necessary pre-processing problems, and rearranges them with their solutions. The pre-processing problems in this paper cover both word-level and phrase/sentence-level problems.

3 PRE-PROCESSING TASKS

This section briefly explains our own English-Korean machine translation (EKMT) system, SmarTran, and presents necessary pre-processing tasks in the translation process. Some pre-processing tasks require corresponding post-processing. We also describe post-processing tasks in this section.

3.1 SmarTran System

Figure 1 shows the logical structure of the SmarTran. Given a source sentence, lexical analysis is done using English lexical dictionary and rules. Necessary lexical information is collected for each word in the sentence. Using the information, sentence is split into several segments for efficient parsing. Each segment is parsed in partial parsing step using idiom recognition and English syntactic rules. Then a global sentence structure is built using the partial parsing results. A transfer is performed on the resulting structure using English-Korean transfer dictionary and transfer rules. The transferred structure is passed to the generation step which generates corresponding target sentence using Korean generation dictionary and rules.

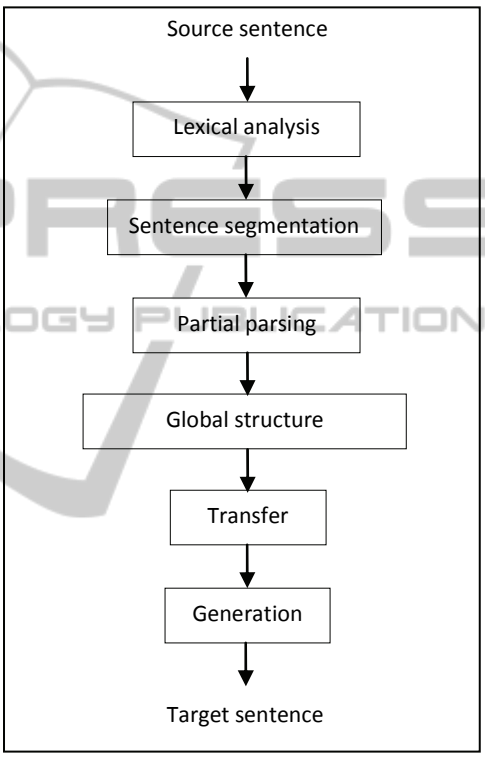

Figure 1: Logical structure of SmarTran system.

3.2 Classification of Pre-processing Tasks

3.2.1 Tasks before Lexical Analysis

Some English sentences include non-word elements such as parentheses, quotation marks, hyphens, list markers, semicolons, and colons. These elements separate a sentence into several translation units. The translation units must be identified before translation process. Also, sentences may include special words such as number (with unit word), composition words, and meaningless words. Some special patterns must be identified to solve differences between source and target languages. We consider 8 pre-processing problems that can be

handled without part-of-speech and other lexical information.

First, a sentence may be split into translation units by semicolon, colon, and bar. The units must be translated independently.

Second, parts of a sentence enclosed by a pair of single or double quotation marks are separate translation units. Parentheses and angle brackets also enclose some parts of a sentence that are also translation units. The enclosed parts must be separated and be translated independently, but they are also elements of other translation unit. Figure 2 shows two examples. The enclosed part Q1 must be translated together with a given sentence as in TU1, while P1 can be translated independently. In case of P1, the target word for TU2 must be identified for post-processing in which the translation result of TU2 is appended to the translation of the target word.

"Next year, I may evaluate it a little closer," said Stan Guest, an uninsured farmer.
TU1: Q1, said Stan Guest, an uninsured farmer
TU2 = Q1: Next year, I may evaluate it a little closer

During the eighth five-year plan period (from 1991 to 1995), the reform successfully completed.
TU1: During the eighth five-year plan period, the reform successfully completed
TU2=P1: from 1991 to 1995

Figure 2: Examples of sentences with translation unit enclosed by double quotation marks.

Third, some sentences have head marks leading list. When the mark is a symbol, it is easily removed. The mark must be recognized and removed when it is a digit or alphabetical digit (ex: i, ii, I, II, …).

Fourth, words abbreviated by apostrophe must be restored for facilitating lexical analysis. For example, "*don't*" must be converted to "*do not*". We build dictionary for such words.

Fifth, words representing numbers must be analyzed to know whether they are ordinal or cardinal. Also, we must identify the combination of number and unit words. In this case, the two words must be combined. For the purpose, we need information about unit and number words.

Sixth, we must identify composite words. Two or more words play a role of a one word noun, verb, adverb, preposition, or conjunction. Composite nouns can be translated by idiom translation method, while composite verbs, adverb, prepositions, and conjunctions can be collected and combined into one word. We need the list of composition words with their translations.

Seventh, some special patterns must be handled. For example, sentences including [~ so that ~] pattern can be rewritten into [~, so ~], and the rewritten sentences are easier to be analyzed. We collect patterns requiring sentence rewriting, and build corresponding rewriting patterns.

Eighth, phrases expressing date must be identified and treated as one word. Also the phrases are translated in separate post-processing for date translation. There are several patterns for representing date. We collect the patterns to be used in identification and build corresponding translation patterns for translation.

3.2.2 Tasks after Lexical Analysis

Some pre-processing problems need lexical information such as part-of-speech, part-of-speech probability, and etc. We present 5 pre-processing tasks as followings.

First, sentences may include phrases expressing human name and his age. The phrases must be combined and treated as one word during lexical and syntactic analysis. It needs corresponding post-processing in which the combined phrases are translated into Korean.

Second, geographical names consisting of pronouns and comma must be combined. For example, in sentence "*I lived in Brynmawr, PA.*", "Brynmawr, PA." is combined, so the phrase can be translated as one word. In order to solve above two problems, we need to know whether a word is pronoun or not.

Third, some sentences start with adverb or adverbial phrase which modifies the following sentence. The modifier can be separated, which can reduce the parsing complexity.

Fourth, some sentences include patterns for which the translation is difficult. Such patterns include [not only ~ but (also) ~], [insist ~ that ~ VERB (base form) ~], [no sooner had ~ than ~], and so on. We need lexical information to match '~' parts in the patterns. For the patterns, we adopt rewriting method using rewriting patterns. In this pre-processing, the sentences matched with the defined patterns are rewritten as directed by the corresponding patterns. The corresponding patterns have compatible meanings and forms that are easier to be analyzed in the rule-based framework. For example, [no sooner had ~ than ~] pattern has [as soon as ~, ~] as its corresponding pattern.

Fifth, we consider comma rewriting for preventing non-constituent segments from occurring by segmentation. For example, in "I need small, fast

computer," the comma can be rewritten into "and," resulting in "I need small and fast computer." This comma rewriting requires information about comma usage (Kim and Park, 2006).

3.2.3 Tasks after Sentence Segmentation

Source sentences with commas can be split by commas resulting in several segments (Kim *et al.*, 2001). There are pre-processing tasks that can be done after sentence segmentation.

First, we search special patterns within each segment. It is difficult to get accurate translation for sentences with such patterns as [so ~ that ~], [it BE-verb ~ ADJ that ~] and [it BE-verb ~ that ~]. The patterns have information about split position and how to combine the parsing results of split sub-segments. We split a segment including such patterns into two sub-segments. Each sub-segment is parsed independently and the parsing results are combined based on the combination rules for the patterns.

Second, there are several patterns for verb "say." When "say" verb has object element enclosed by a pair of double quotation marks, the order of words in the sentence may be different from normal sentences. It is difficult to parse such sentences in rule-based framework, so sentence elements repositioning is required. Figure 3 shows two examples. The first example is from figure 2. After translation units are identified, the units are rearranged. Actually, sentence elements repositioning is a rearrangement of translation units. This repositioning results in a rearranged sentence in which order of words was described by the existing syntactic rules. Through the repositioning, we identify the role of the segments and this information is used to generate global sentence structure from partial parsing results of each segment.

"Next year, I may evaluate it a little closer," said Stan Guest, an uninsured farmer.
⇨ Q1, said Stan Guest, an uninsured farmer
⇨ Stan Guest, an uninsured farmer, said, Q1.
"We continue to believe the position we've taken is reasonable," a Morgan Stanley official said.
⇨ Q1, a Morgan Stanley official said.
⇨ A Morgan Stanley official said, Q1

Figure 3: Examples of sentences elements repositioning.

Third, some phrases or clauses can be inserted which play roles of adverb, modifier, and etc. The insertion is generally separated by commas. So the comma-separated insertion patterns can be a pre-processing target. Figure 4 shows examples. In the first sentence, "if it ~ strong dollar" segment is an inserted subordinate clause (INS_SB) and can be extracted and handled independently. A segment "Donald Taffner" from the second sentence is an appositive of the preceding word "agent". The segment is extracted and translated separately, while "Thames's U.S. marketing agent is preparing to do just that" is translated as one translation unit. In third sentence, sentential modifier "in fact" is extracted and the translation result is appended to the target sentence. The identification and processing of the insertion patterns can facilitate the parsing and improve the quality of target sentence.

A widening of the deficit, if it were combined with a stubbornly strong dollar, would exacerbate trade problems.
INS_SB: if it were combined with a stubbornly strong dollar
Thames's U.S. marketing agent, Donald Taffner , is preparing to do just that.
INS_APP: Donald Taffner
Radio Free Europe, in fact, is in danger of suffering from its success.
INS_MOD: in fact

Figure 4: Examples of insertion patterns.

3.3 Post-processing Tasks

We explain post-processing tasks required by some pre-processing tasks above described.

First, translation results of the translation units split by non-word elements are combined using the split element. The order of translation units must be kept and split elements are inserted into the combination positions. This post-processing locates after the generation step in figure 1.

Second, enclosed parts by quotation marks or parentheses are translated and then the results are appended to the translation of the target words. In post-processing, we search target words in the resulting parse tree after the transfer steps in figure 1.

Third, we have to translate the combined date and name-age words. This translation is based on the translation patterns as in figure 5. This post-processing is performed in transfer step in figure 1.

Fourth, segments with special patterns in section 3.2.3 are split according to their split information. Also the patterns have information about how to combine the parsing results of the split segments. In this post-processing, we combine the parsing results into one global sentence structure based on the combination rules described in the patterns. That is,

the post-processing is done after the partial parsing in figure 1. Figure 6 shows examples of special patterns consisting of 3 parts. The first part is a sentence pattern, the second gives information how to split the segment, and the third means how to combine the parsing results. In the example, '+' means combine two trees and '1_SUBJ_2' means the first tree's subject is the second tree. In each pattern, the first and second parts are for pre-processing and the last part is for post-processing.

[Month NUM1, NUM2] → [NUM2 년 Month 월 Num1 일]: January 1, 1998
[Month, NUM] → [NUM 년 Month 월]: January, 1998
[NUM, Month] → [Month 월 NUM 일]: 1, January
[PRONOUN, NUM] → [NUM 살인 PRONOUN]
Antonio L. Savoca , 66 *, was named president*

Figure 5: Examples of translation patterns for data and name-age.

1. [~ so A that B], ([~ so A] , [and B]) , +;
2. [it BE-verb A ADJ that B], ([it BE-verb A ADJ], [B]), 1_SUBJ_2;
3. [it BE-verb A that B], ([it BE-verb A], [B]), 1_SUBJ_2;

Figure 6: Examples of special patterns within a segment.

Fifth, during the sentence elements repositioning in section 3.2.3, we identify the role of the segments. In post-processing, we generate a global sentence structure by combining partial parsing results as directed by the information about role of segments.

Sixth, the parsing tree of the insertion segment is added to the global sentence structure according to the insertion types. For example, a tree for INS_SB segment is added as subordinate clause, a tree for INS_APP is added as appositive of the target word, and a tree for INS_MOD is added as sentential modifier. This post-processing can be performed after constructing global sentence structure.

4 EXPERIMENTS

In this section, we show the usefulness of the defined pre-processing tasks. For the purpose, we search sentences with the defined pre-processing problems. We use sentences from 4 domains in Penn Treebank corpus: WSJ (Wall Street Journal), Brown, ECTB (English-Chinese Tree Bank), and IBM. We have 53,838 sentences from WSJ, 50,440 sentences from Brown, 3,825 sentences from ECTB, and 4,404 sentences from IBM. We do not run the SmarTran system, and only search the pre-processing problems defined in section 3. From table 1 to table 3, we summarize the defined pre-processing problems. And the following three tables present the statistics for the sentences with the problems.

Table 1: Pre-processing tasks before lexical analysis.

Task ID	Description	Target patterns
1	Separation by non-word elements	semicolon, colon, bar
2	Separation by non-word elements	'', "", (), <>
3	Head leading list	I, II, …, 1), 2), …,
4	Number + unit	$, %, , dollar, cm, …
5	Composition words	
6	Special patterns	[~ so that ~], …
7	Date patterns	[January NUMBER], …

Table 2: Pre-processing tasks after lexical analysis.

Task ID	Description	Target patterns
1	Human name + age	[PR-NOUN + NUMBER]
2	Geographical name	[PR-NOUN + , + PR-NOUN]
3	Head ADV, AVP	42 adverbs, 4 adverbial phrases
4	Special patterns	[not only ~ but (also) ~], [insist-like verbs that ~ VERB (base form) ~], …
5	Comma rewriting	[ADJ(COMPR), ADJ(COMPR)], [PR-NOUN(HYPHEN), PR-NOUN(HYPHEN)], …

Table 3: Pre-processing tasks after sentence segmentation.

Task ID	Description	Target patterns
1	Special patterns	[~ so (such) ~ that ~], [too ~ TO_INF ~], …
2	Say-like verbs	say(said), tell(told), ask(asked), explain(explained), …

Table 4: Statistics for sentences requiring pre-processing before lexical analysis (%).

Task ID	WSJ	Brown	ECTB	IBM
1	6.7	11.1	4.7	0
2	20.1	19.7	13.3	8.3
3	0.4	0.4	0.2	0.2
4	13.1	1.0	1.5	0
5	5.2	5.6	9.1	2.7
6	0.6	1.1	0.7	0.5
7	5.7	1.7	4.7	0
	51.8	40.6	34.2	11.7

Table 5: Statistics for sentences requiring pre-processing after lexical analysis (%).

Task ID	WSJ	Brown	ECTB	IBM
1	0.6	0.4	0.5	0
2	7.1	3.2	8.2	0.4
3	9.7	8.8	3.3	2.5
4	0.2	0.4	0.5	0
5	0.2	0.1	0.1	0
	17.7	13.0	12.7	3.0

Table 6: Statistics for sentences requiring pre-processing after sentence segmentation (%).

Task ID	WSJ	Brown	ECTB	IBM
1	1.4	3.3	2.0	0.5
2	27.7	13.6	13.2	0.9
	29.1	16.9	15.3	1.4

From the above three tables, we know that there are many sentences which have the pre-processing problems. The SmarTran can generate better translation results using the pre-processing tasks. Most sentences from WSJ can be benefited from the pre-processing tasks. About 78% sentences from Brown and about 60% from ECTB are also benefited. Therefore, the proposed pre-processing tasks are expected to improve the translation quality.

5 CONCLUSIONS

This paper presents required pre-processing tasks and corresponding post-processing tasks in English-Korean machine translation. The pre-processing has purpose of solving differences between English and Korean and facilitating the analysis of sentences including non-word elements. Also, we classify the tasks based on the time when the tasks should be done. This classification augments the structure of the existing EKMT system. For pre-processing problems, we present solutions such as sentence split, symbol or words deletion, word conversion, combination of words, rewriting (words, phrases, and comma), segment removal from the segment list which is from sentence segmentation step, and sentence elements repositioning.

Some of pre-processing solutions are already developed, and others are being studied. We need a representation method of patterns and other information. Also, we must verify the solutions with many examples. Some methods may cause side effects, so we need solution to avoid them. Further, we must test EKMT system with proposed pre- and post-processing with many sentences and measure how much the translation quality is improved. We expect the pre-processing tasks will improve the translation quality.

ACKNOWLEDGEMENTS

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(KRF) funded by the Ministry of Education, Science and Technology(2010-0010815).

REFERENCES

Kim S.-D. 2008. Study on Sentence Rewriting in English-Korean Machine Translation. *Proceedings of Korean Computer Conference*. In Korean.

Kim S.-D., B.-T. Zhang, and Y. T. Kim. 2001. Learning-based Intrasentence Segmentation for Efficient Translation of Long Sentences. *Machine Translation*, 16(3), 151-174.

Kim S.-D. and S.-H. Park. 2006. Comma Usage Classification for Improving Parsing Accuracy of Long Sentences in English-Korean Machine Translation. *Proceedings of Korean Information Science Society.* In Korean.

Kim Y.-S. and Y. T. Kim. 1998. Semantic Implementation based on Extended Idiom for English to Korean Machine Translation. *Journal of the Asia-Pacific Association for Machine Translation*, 20, 23-39.

Yoon S. H. 1993. Idiom-Based Efficient Parsing for English-Korean Machine Translation. *Ph.D Dissertation of Seoul National University.* In Korean.

Yuh S. H., H. M. Jung, Y. S. Cahe, T.W. Kim and D.-I. Park. A Preprocessor for Practical English-to-Korean Machine Translation. 1996, *Proceedings of Korea Language Engineering Research Society*, 313-321. In Korean.

Yuh S. H., H. M. Jung, T. W. Kim, D.-I. Park and J. Y. Seo. 1997. Preprocessing of Hyphenated Words for English-Korean Machine Translation. *Proceedings of Korean Information Science Society*, 24(2). 173-176. In Korean.