For the Proper Treatment of Long Sentences in a Sentence Pattern-based English-Korean MT System

Yoon-Hyung Roh, Munpyo Hong, Sung-Kwon Choi, Ki-Young Lee, Sang-Kyu Park

Speech and Language Information Research Center, Electronics and Telecommunications Research Institute (ETRI)
Daejon, Korea
{yhnoh, hmp63108, choisk, leeky, parksk}@etri.re.kr

Abstract

This paper describes a sentence pattern-based English-Korean machine translation system backed up by a rule-based module as a solution to the translation of long sentences. A rule-based English-Korean MT system typically suffers from low translation accuracy for long sentences due to poor parsing performance. In the proposed method we only use chunking information on the phrase-level of the parse result (i.e. NP, PP, and AP). By applying a sentence pattern directly to a chunking result, the high performance of analysis and a good quality of translation are expected. The parsing efficiency problem in the traditional RBMT approach is resolved by sentence partitioning, which is generally assumed to have many problems. However, we will show that the sentence partitioning has little side effect, if any, in our approach, because we use only the chunking results for the transfer. The coverage problem of a pattern-based method is overcome by applying sentence pattern matching recursively to the sub-sentences of the input sentence, in case there is no exact matching pattern to the input sentence.

1 Introduction

More than ten years have already passed since the first commercial English-Korean MT system came to the market in Korea. However, the translation quality of the best systems, most of which adopt the RBMT approach, has stagnated at around 60~65%. The systems especially show poor performance for long sentences containing more than 25 words. For practical use of the MT systems the notable improvement of the translation performance is demanded especially for long sentences (Park and Oh, 1999). In our opinion, the reason that the translation performance of most English-Korean MT systems has come to a standstill is that they adopt a rule-based approach and there has been no breakthrough in the treatment of long sentences.

To deal with the issues, ETRI has pursued a so-called Sentence Pattern-based MT (henceforth SPBMT) since 1999 (Seo at al., 2001). The sentence pattern is a phrasal chunk pattern with translation information for a sentence. The main idea of SPBMT is to shift the translation load from the analysis (the engine) to the pattern (the data), as is often the case in other data-driven approaches. However, the SPBMT in its original form suffers from a serious data sparseness problem.

The most widely adopted solution to the problems of long sentences is sentence partitioning (Kim and Ehara, 1994; Kim and Kim, 1995; Li et al., 1990). But the partitioning methods show limits in accuracy, and may lead to the errors in the subsequent analysis because a sentence is partitioned usually without a deep analysis.

This paper describes our research into the treatment of long sentences in the SPBMT paradigm. Our strategy is to divide the analysis units into two levels, i.e. phrase level and clause level, in order to minimize the side effect of the sentence partitioning, and to reduce the ambiguities of a rule-based module by employing the sentence patterns.

The typical problems of the RBMT and the SPBMT approach in dealing with long sentences will be sketched in the following section. In section 3, the issue of sentence partitioning as a typical method for processing long sentences will be tackled. In section 4, a new sentence pattern-based translation method for long sentence translation will be introduced. The results of the experiments to assess the proposed method will be discussed in section 5. Finally in section 6 we will conclude our discussion with some words about remaining work.

2 Translation of Long Sentences

Table1 shows the translation quality of a highly rated commercial English-Korean MT system in Korea:

Sentence Length (number of words)	**~20**	**20~30**	**31~**
Translation Accuracy	68.1%	63.1%	55.0%

Table 1: Translation Quality of a commercial English-Korean MT System

As for the sentences with less than 20 words, the system shows a moderate translation accuracy (68.1%). However, the translation accuracy drops drastically as the number of words in the sentences surpasses 30 (55.0%). In this chapter, we investigate the problems of long sentence translation from the viewpoint of the RBMT and the SPBMT, and discuss a method that can complement the problems of each approach.

2.1 Parsing of Long Sentences

One of the deficiencies a general RBMT system shows is that a lot of ambiguities are generated during the parsing, because the analysis is performed by applying the small number of rules repeatedly. Especially the system speed and analysis accuracy drop drastically because of the explosively increasing ambiguity as a sentence becomes longer. Table 2 shows the analysis results of an average performance chart parser in proportion to the sentence length.

Sentence Length	**20**	**30**	**40**	**50**
Active Node#	104,900	514,200	1,700,000	3,678,000
Parsing Time(s)	0.21	2.29	7.82	21.0

Table 2: Chart Parsing Results in proportion to the Sentence Length

Although the correlation between the sentence length and the ambiguity is clear, we can also find out that the great portion of ambiguities are caused by the wide-range analysis rules (i.e. the rules for the higher nodes in a syntactic tree). On the contrary, the narrow-range rules, say a rule for phrase node such as NP, do not usually generate an ambiguity as a sentence becomes longer (Abney, 1996). In other words, as the nodes ascend in a syntactic tree, they tend to be more ambiguous and less correct. As the applicable combinations of the rules become large, the discrimination power of rules decreases. For this reason, our approach does not rely on the structural information of higher nodes than the phrase level in a syntactic tree.

Concerning parsing efficiency, it takes about 8 seconds for the sentences with 40 words, and 20 seconds for 50 words sentences. In case of the traditional chart parsing method, the processing time increases cubic times faster than the sentence length. For this reason, a special treatment of long sentences is needed in most practical MT systems.

2.2 Translation of Long Sentences by Patterns

To resolve the above-mentioned problems in parsing, large-scale patterns of sentence range are built in the SPBMT approach, and the analysis is performed on the basis of the patterns. If there is an exact matching pattern for the input sentence, high quality translation can be expected compared with the RBMT (Seo at al., 2001). However, it takes enormous time and effort to build high-quality translation patterns in a large scale, and it is rather unrealistic to build enough sentence patterns to cover not only relatively short sentences but also longer sentences[1].

[1] Although the automatic construction of sentence patterns is not impossible in the current framework, the automatically constructed patterns must be revised by skilled translators and linguists.

3 Sentence Partitioning

An applicable method for long sentence processing is the sentence partitioning. Parsing speed and performance can be improved by partitioning a sentence into smaller units before the parsing of the sentence. But it can produce a lot of side effects when sentence partitioning is carried out with only limited context and analysis information.

Typical errors in the sentence partitioning can be summed up as follows.

1) Detection Error of Clause Boundaries

i) "The night their team defeated Russia, Kyoko Ebata, /28, a Tokyo artist, was out with friends in a local bar."[2]

ii)"The last time I talked to the president about the budget /and the appropriations bills was the middle of July."

2) Detection Errors of Clause Hierarchies

iii) He said he was told that an interrogator would use the tires to stand on, /while water was poured into the room and the prisoner electrocuted.

3) Detection Errors due to inserted clauses

iv) They want to, /when the political order is given, bring these men in by air.

Generally, even if the boundary point between clauses is recognized correctly, if the partitioning is done as 2), it may cause translation errors because of the wrong hierarchy in the clausal structure. However, such errors are hard to correct without analyzing the entire sentence. The dilemma is that it is very difficult to partition a sentence without recourse to deep syntactic analysis in the partitioning step which usually precedes the syntactic analysis. But no problem occurs in such cases as ii) and iii), if we use only the chunking information on the phrase level of the parse tree.

Many of the partitioning errors have to do rather with the clausal structure than with phrasal structure. Therefore we try to carry out long sentence translations by separating the processing units into the phrasal unit and the clausal unit.

4 Sentence Pattern-based English-Korean MT for Long Sentences

In this chapter, we would like to take a closer look at the sentence pattern-based system for long sentence translations.

4.1 System Overview

Figure 1 sketches the whole configuration of our new SPBMT system for long sentences. After the chunking and tagging of an input sentence, the sentence length is checked[3]. The definition of a "long" sentence can be made variously, however we take a sentence for long, if the number of the words (after chunking of frozen expressions like "September 11th.") surpasses a threshold. We plan to improve the measure by applying weight to each part-of-speech because each part-of-speech makes a different contribution to parsing ambiguity. If it turns out to be a long sentence, the sentence partitioning is carried out. Subsequently, each partition is parsed and only phrasal chunk structure is extracted from parse tree. Then, translation is performed with a sentence pattern, if pattern matching succeeds. If the pattern matching fails, clausal structure is analysed with phrase patterns called slot patterns in our approach. Finally partial sentence translation is performed according to the analysed clausal structure.

4.2 Chunking & Tagging

Chunking targets mainly proper nouns, time adverbs, and lexically fixed expressions with the purpose of reducing the sentence length and increasing the tagging performance. The tagging module employs the N-gram model reflecting lexical information and passes best 2 candidates for tagging accuracy and parsing efficiency.

[2] The "/" symbol denotes the partition boundaries detected by the sentence partitioning module.

[3] Note that the term "chunking" in this context is meant as a pre-processing module for a multi word expressions for time and locations, e.g. September 11th. Seoul, Korea, etc. Unless otherwise indicated, the term "chunking" as in "chunking structure" means building a phrase-structure like NP, PP, AP.

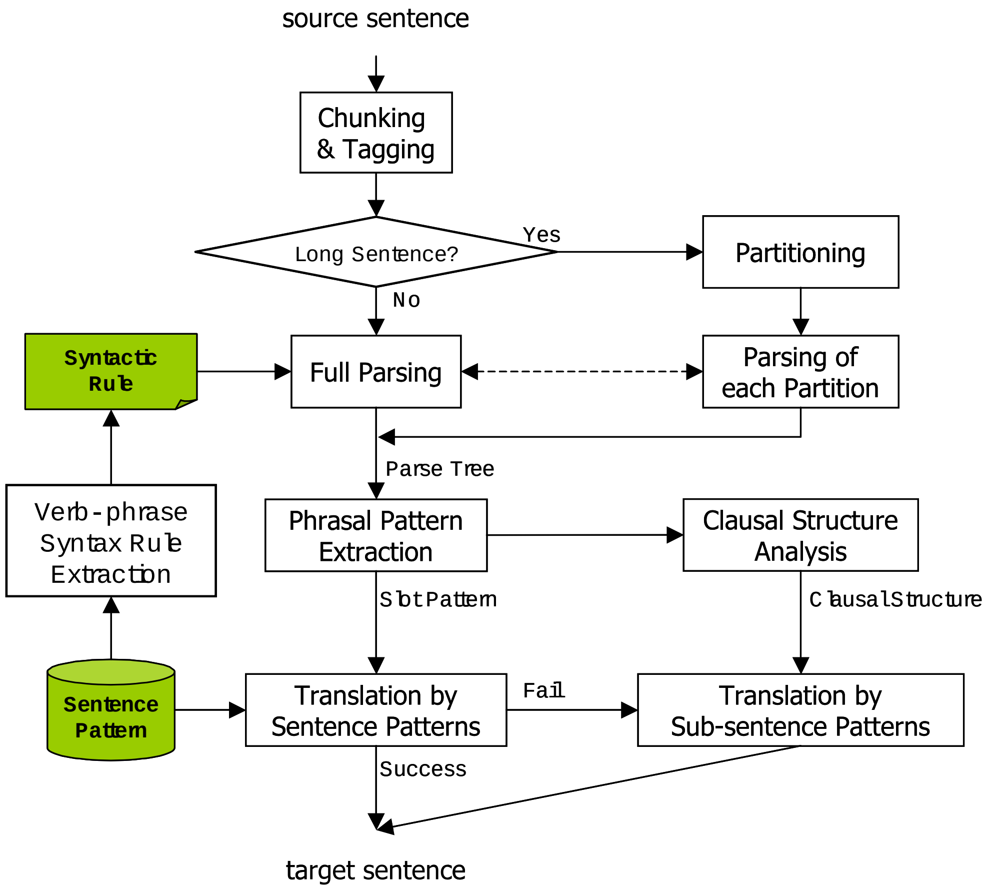
source sentence
Chunking & Tagging
Long Sentence?
Yes
No
Partitioning
Syntactic Rule
Full Parsing
Parsing of each Partition
Parse Tree
Verb-phrase Syntax Rule Extraction
Phrasal Pattern Extraction
Clausal Structure Analysis
SbtPattern
ClausalStructure
Sentence Pattern
Translation by Sentence Patterns
Fail
Translation by Sub-sentence Patterns
Success
target sentence

Figure 1: Sentence Pattern-based English-Korean MT for Long Sentences

4.3 Partitioning

If the chunked input sentence turns out to be a long sentence, the sentence is partitioned.

The partitioning process is as follows.

1) Partitioning candidates are selected by the syntactic clue with priority such as sentence punctuation, a conjunction, a relative, an interrogative, etc.
2) Final 2~3 of candidates are left by checking some conditions such as the existence of the main verb, the partition length, etc.
3) Each partition according to each partitioning candidate point is parsed.
4) By considering the parsing score on each candidate point, the most appropriate partitioning point is selected.

We expect that in the case of wrong partitioning point such as a point in an included clause, the score of parse tree of the partition will be somewhat low, and the point is not taken into account any more.

The reason for the partitioning with only one point is to minimize the side effect of partitioning. An experiment showed that there is a considerable improvement in terms of the system speed even with only one point partitioning. The following example shows partitioning candidate points:

[Input Sentence]: *"We're told to look for an announcement under which the Russians would temporarily participate in the NATO command structure while the political leaders, including the* *two presidents when they speak today, try to work out the arrangements for a much broader Russian participation in the peacekeeping force."*

[Partitioning candidate points]: … *in the NATO command structure /while the political leaders, including the two presidents /when they speak today, try to ....*

4.4 Full Parsing

Although a sentence pattern is a phrase-level pattern for which a partial parsing seems to be enough, a full parsing is conducted because of the following reasons:

Firstly, even if a partial parsing may be sufficient to extract the slot patterns, a partial parsing fails to reflect wide-range context or a dependency. For example, the type of verb is decisive for the recognition of a phrase, and the linking of a subject and its main verb is an important clue for the structural disambiguation. More accurate results are expected by the consideration of the wider range context and deep analysis in comparison with a partial parsing.

Secondly, many sentence-range rules can be generated from existing sentence patterns. We can obtain more accurate and cohesive parsing results by having sentence-range patterns reflected in the parsing rules.

The following are the partitions for the parsing of the example sentence:

[Partitions for Parsing]
while: *(We're told to look for ... NATO command structure) (while the political leaders, including the two presidents when they speak today, try to ... the peacekeeping force.)*
when: *(We're told to look for ... NATO command structure while the political leaders, including the two presidents) (when they speak today, try to ... in the peacekeeping force.)*

In case of 'when', because the partition *"We're told to look for an announcement under which the Russians would temporarily participate in the NATO command structure while the political leaders, including the two presidents"* is an abnormal sentence, 'when' is excluded from the partition points by parsing score.

The following are the parsing results of the partitions of the example sentence:

[Parsing Results of the Partitions]
(S (NP We) (VP 're (VP told (TOINF (VP to (VP look_for (NP an announcement) (PP under)))))) (SBAR (WHNP which) (SS (NP the Russians) (VP would temporarily (VP participate (PP in (NP the NATO command structure)))))))
(S (NP (NP the political leaders) -COMMA- (PP including (NP (NP the two presidents) (SBAR (WHADVP when) (SS (NP they) (VP speak today))))) -COMMA-) (VP try (TOINF to (VP work_out) (NP the arrangements) (PP for (NP (NP a (ADJP much broader) Russian participation) (PP in (NP the peacekeeping_force)))))))

4.5 Phrasal Pattern Extraction from a Parse Tree

The phrasal pattern is extracted by recognizing the chunking range of phrases subcategorized by a verb such as NP, AP, PP, etc. from the parse tree. We call the resulted phrasal pattern a slot pattern (Seo at al., 2001).

The followings are the chunking result extracted from parse tree and slot pattern of the example sentence:

[Extracted Phrasal Chunk]
(NP We) 're told (IPREP to) look_for (NP an announcement) (IPREP under) which (NP the Russians) would temporarily participate (PP in the NATO command structure)
(NP the political leaders) -COMMA- (PP including the two presidents) when (NP they) speak today -COMMA- try (IPREP to) work_out (NP the arrangements) (PP for a much broader Russian participation in the peacekeeping_force)
[Slot Pattern]: nViVniCnVpCnTpCnVTViVnp

4.6 Translation by Sentence Patterns

A sentence pattern consists of a condition part and a transfer part. If the condition is satisfied, transfer is conducted.

The following is a sentence pattern example:

{NP1 VERB1!:[(etype == [x3]) _AND (eform == [vb]) _AND (voice == [passive])] IPREP1:[eroot == [to]] VERB2:[(etype == [t1]) _AND (eform == [vb]) _AND (voice == [active])] NP2 } -> { NP1:[kcase := [topic]] NP2:[kcase :: VERB2.kflex2, kjcode :: VERB2.kfcode2, kcase :: [obj]] VERB2 IPREP1:[kroot :: VERB1.kflex3, kcode :: VERB1.kfcode3, kroot :: [□_□□], kcode :: [ej00202]] VERB1! }

4.7 Clausal Structure Analysis

Clausal structure analysis is carried out by recognizing ranges of each clause first, and analyzing the relations between each clause. Ranges of a clause are recognized by searching for the starting point candidate of each clause first, then recognizing all possible ending points of each starting point, checking some condition such as the existence of a main verb. The clausal structure is analyzed by carrying out traditional parsing on the recognized simple clauses.

The following is the result of clausal structure analysis of the example sentence:

[Result of Clausal Structure Analysis]
(nViVniC((nVp)C(nT(pC(nV))TViVnp)))

4.8 Translation by Sub-sentence Patterns

When the clausal structure is analyzed, translation is conducted according to the ranges in the clausal structure in the top-down manner.

As the root of the analysis tree denotes the whole sentence range, sentence pattern-based translation is tried to the sub-clauses in the child nodes. When the sub-clauses are translated, the final translation result will be produced by matching sentence patterns to the reduced slot pattern. In case there is no exact matching sentence pattern for a sub-clause, the sub-clauses corresponding to its child nodes will be detected and translated in a recursive manner.

The following shows translation by sub-sentence pattern on the example sentence:

[Translation by Sub-sentence Patterns]

```
nViVniC((nVp)C(nT(pC(nV))TViVnp)) -> s (fail)
  1.1 (nVp)C(nT(pC(nV))TviVnp -> s (fail)
     1.1.1 nVp -> s (success)
     1.1.2 nT(pC(nV))TviVp -> s (fail)
        1.1.2.1 pC(nV) ->p (success)
        1.1.2.2 nTpTViVp ->s (success)
     1.1.3 sCs -> (success)
  1.2 nViVniCs -> s (success)
```

5 Experiments

To assess the proposed method, we conducted three experiments concerning partitioning, the clausal structure analysis and the translation quality.

5.1 Partitioning

120 sentences with about 40 words on average were used as test corpus. The partitioning error, the chunking error, and parsing time are evaluated on both original sentence and partitioned sentence. In the case of 16 sentences, a partitioning candidate was incorrectly detected, because the sentences consisted of only one clause or a relative clause.

The following are the number of errors and parsing time related to partitioning.

	Partitioning	Chunking	Parsing Time
Whole Sentence		51	630s
Partitions	14	48	85.6s

Table 3: Experimental Results on the Partitioning

There were 14 partitioning errors, and the number of the chunking errors did not increase as we adopted the partitioning method. Parsing time was shortened to about 1/7. To sum up, chunking accuracy was not affected by the partitioning, and the parsing time decreased enormously. This shows that our approach is an effective way to overcome the limits of partitioning.

5.2 Clausal Structure Analysis

We experimented on clausal structure analysis of the above 104 sentences by comparing 1) the clausal structure analysis by parsing as described in 4.4, and 2) the clausal structure analysis by chunking and clausal structure analysis as described in 4.5 and 4.7. In method 1), the clausal structure is extracted from the parse tree. The number of errors is 36 in method 1), 17 in method 2), so method 2) outperforms method 1), where the number of errors in method 1) includes the partitioning errors.

5.3 Translation Accuracy

Translation accuracy was assessed with 100 test sentences. The test corpus was mostly news scripts on the web and some interview articles on a general domain. The average length of the sentences was 23.6 words. The translations were evaluated according to the following scoring criteria:

4: Perfect Translation
3: Meaning of the sentence conveyed, however, naturalness is lacking
2: Partial phrase level translation
1: Partial word level translation
0: No translation

We evaluated the translation accuracy of SPBMT in its original form and the new method. There was a performance enhancement of 5.2% by adopting the new method.

	Total Score	Translation Accuracy
SPBMT	254	63.5%
Proposed Method	275	68.75%

Table 4: Translation Accuracy of SPBMT and Proposed Method

6 Conclusion

This paper has described an effective pattern-based method to deal with long sentences. As a sentence gets longer, many problems are encountered such as system and coverage problem. The traditional sentence partitioning method to tackle these problems has a limit in its performance.

We presented a method to solve the problem by separating phrasal chunking and clausal structure analysis using the characteristics of a rule-base method and a pattern-based method. There was about 5.2% performance enhancement in translation accuracy.

For future work, we will investigate the adequate number of partitioning points and expand partitioning targets. Also, we will enhance the performance of clausal structure analysis by coordinate structure analysis.

7 Bibliographical References

Steven Abney, 1996, Part-of-speech tagging and partial parsing. *Corpus-Based Methods in Language and Speech*. Kluwer Academic Publishers.

Yeun-Bae Kim and Terimasa Ehara, 1994, A Method for Partitioning of Long Japanese Sentences with Subject Resolution in J/E Machine Translation, In *Proceedings of the 1994 ICCPOL*.

Sung Dong Kim and Yung Taek Kim, 1995, Sentence Analysis using Pattern Matching in English-Korean Machine Translation, In *Proceedings of the 1995 ICCPOL*, pp. 199-206.

Wei-Chuan Li, Tzusheng Pei, Bing-Huang Lee, and Chuei-Feng Chiou, 1990, Parsing Long English Sentences with Pattern Rules. In *Proceedings of 25th Conference of COLING*, pp. 410-412

Se-Young Park and Gil-Rok Oh, 1999, Machine Translation in Korea, In *Proceedings of MT-Summit 1999*, pp. 100-104.

Young-Ae Seo, Yoon-Hyung Roh, Ki-Young Lee, Sang-Kyu Park, 2001, CaptionEye/EK: English-to-Korean Caption Translation System Using the Sentence Pattern, In *Proceedings of MT-Summit 2001,* pp. 325-329.