

# Long Sentence Partitioning using Structure Analysis for Machine Translation

Yoon-Hyung Roh, Young-Ae Seo, Ki-Young Lee, Sung-Kwon Choi

Linguistic Engineering Department

Electronics and Telecommunications Research Institute

161 Kajong-dong, Yusong-gu, Taejon, 305-350, Korea

{yhnoh, yaseo, kylee, choisk}@etri.re.kr

## Abstract

in machine translation, long sentences are usually assumed to be difficult to treat. The main reason is the syntactic ambiguity which increases explosively as a sentence become longer. Especially, in the machine translation using sentence patterns, a long sentence causes a critical coverage problem. In this paper, we present a method of sentence partitioning which recognizes sub-sentence ranges by structure analysis, reducing the length of a sentence for translation. For the analysis of the clausal structure, phrase-level sentence patterns which have only a little syntactic ambiguities are employed. The structure analysis is conducted by the recognition of starting points of all clauses, dependency analysis, and depth analysis. Then, the ranges of sub-sentences are extracted based on the depth by stages. Our method was evaluated on 108 sentences extracted from CNN transcripts. It showed 85.2% accuracy in the detection of simple sentences.

## 1 Introduction

In machine translation, the quality and the efficiency of translation of long sentences is very low. It is because as the length of a sentence goes up, the syntactic ambiguity of sentence increases rapidly. Also, in the machine translation using the sentence pattern, we can obtain a high translation quality, but as the length of input sentence increases, critical coverage problem is encountered[Seo et al., 2001]. For these problems, a long sentence should be partitioned

into smaller fragments and translated with smaller units. The enlargement of a sentence is usually due to the embeddings and and conjunctions of clauses. So, the partitioning of a long sentence is achieved by recognizing the relative clauses and the conjunctive clauses.

This paper presents a method of recognizing the ranges of sub-sentences in a sentence by the clausal structure analysis using sentence pattern. A sentence pattern is a phrase-level pattern of a sentence which doesn't have much syntactic ambiguity. So, we analyze the syntactic ambiguities in clause-level with the whole sentence pattern information. We get the inspiration from the fact that a long sentence is usually formed by conjunctions of clauses, so the traditional phrasal rule is not sufficient for resolving the dependencies in wide range between clauses. The clause-level structure can be analyzed effectively by considering global sentence pattern.

The sentence structure analysis is to determine the ranges of sub-sentences, the relations between them, and the depth of each clause. And the sequence of ranges of sub-sentences are extracted based on the depth. The goal of this paper is not the perfect analysis of the clausal structure, but the determination of the sub-sentence ranges for the translation. The partition ranges are only a part of sub-sentence ranges resulted from structure analysis, and the partition ranges are mainly used for the pattern-based translation system on which this paper is based.

## 2 Related Work

In syntactic analysis, because of the difficulty of long sentence analysis, a lot of researches about the coordinate conjunction or the sentence segmentation were made. [Peh and Ting, 1996]

segments a sentence by disambiguating the role of link words such as comma and conjunction using a neural network model. In [Kim and Kim, 1997], a method of sentence segmentation by the segmentation point and functionality checking was presented. These methods partition a sentence only by the keyword corresponding to clues of segmentation and the context in the vicinity. And the dependency analysis of clauses is little accomplished, though the clauses in a sentence have a structure.

In [Kim and Ehara, 1994], partition points are searched by using multi-layered pattern matching. In [Okumura and Muraki, 1990], [Kurohashi and Nagao, 1994] and [Yoon and Song, 1997], they analyze coordinate structures using a parallelism. Similarly, we analyze the coordination using a parallelism. But instead of thorough analysis of coordination, we conduct simple analysis of coordination only between clauses, and leave the remaining ambiguities to be resolved by the translation patterns[Seo et al., 2001].

### 3 Machine Translation using the Sentence Patterns

In spite of technological attempts for many years, the translation quality of different language pairs such as English-to-Korean is very low. The major causes would be enumerated as follows [Choi et al., 1999]:

- Many ambiguities is caused by uncertain boundary of right association in parsing
- Incorrect translation in whole sentence is occurred by not considering global translation pattern such as sentence, but local translation patterns such as phrases or clauses as translation unit of sentence.
- Translation quality comes to a standstill due to conflicts in massive translation rules accumulated every year.

To solve these problems, new machine translation methodology based on the sentence pattern has been developed. According to this methodology, CaptionEye/EK, an English-to-Korean caption translation system, has been developed. The new machine translation methodology has characteristics of both shallow bottom-up parsing by protectors and top-down matching by structure-oriented sentence patterns. The protector is a special

part-of-speech such as a verb or a conjunction which cause many ambiguities in a sentence analysis.

CaptionEye/EK has basically the formalism as follows: the system carries out English sentence analysis in which English sentence pattern is built by partial parser between protectors, tries to match English sentence pattern with its Korean sentence pattern, and then generates a Korean sentence from it. With the pattern based methodology, we can reduce the syntactic ambiguities by confining the range of partial parsing, and prevent the unlimited generation of target translation by matching English sentence pattern with its Korean sentence pattern. But as the length of sentence increases, the number of sentence pattern to build increases explosively, causing serious coverage problem. For this, a long sentence should be partitioned into smaller fragments and translated with smaller units.

## 4 Sentence Structure Analysis

The overall process of sentence structure analysis is shown in Figure 1. First, a sentence pattern for the structure analysis is generated. The sentence pattern makes it easy to use adequate information of whole sentence pattern for the clause-level structure analysis and to describe the context for the recognition of omitted conjunction. Then, the starting points of all clauses are recognized. Third, the dependency between each clause is analyzed. Finally, the depths of all clauses are determined. Once the depths are determined, the partitioning ranges are extracted based on the depth.

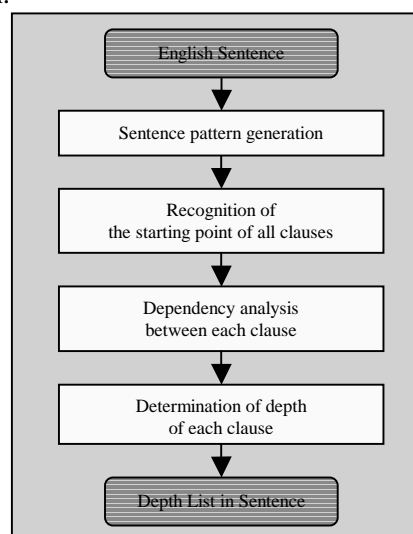


Figure 1. Overall Process of Sentence Structure Analysis



example, in the sentence “They've been told that they've been told that they expect Yugoslav army soldiers and also special Serbian police, the so-called MUP, their instructions on dealing with the Yugoslav forces is the same. ”, there is ambiguity where the third clause begins, and of the noun phrases between two clauses the last noun phrase, “their instruction” is considered to be partitioning point.

In recognizing starting points, we estimate the main verb and the noun phrase corresponding to subject. We select as a main verb the first verb with tense, which is not an infinitive or a participle. Then, we recognize the first noun phrase before the main verb as the subject. The following example shows recognized starting points of the sentence in section 4.1, where ‘/’ means partitioning point and underlined verbs are main verbs.

#### [Input Sentence]

A friend /who was with him said /that he was concerned about the media coverage and the controversy /that has resulted from those crosses.

#### [Recognition of Starting Points]

n/CVpV/CnVnCn/CVp

### 4.3 Dependency Analysis

The dependency is analyzed by the rule for dependency resolution and by the parallelism analysis. The dependency is described by dependency link and dependency relation. The dependency link represents a point which a clause is subordinate to or coordinate with, and the dependency relation represents whether a clause is subordinate to other clause or coordinate with other clause.

If a clause ‘A’ depends on the other clause ‘B’ subordinately, the dependency link of the starting point ‘A’ points to the starting point ‘B’, and dependency relation is subordinate. A slot designates the phrase or the part-of-speech composing sentence pattern. For example, in the sentence “I also know that many churches and other religious institutions have been taking up collections, we are grateful for that as well”, the clause beginning with “many churches” is subordinate to the main clause of the sentence, and the clause beginning with “we are” is coordinate with the main clause.

The type of rule for dependency resolution is classified according to the type of conjunction or

clause as follows. (In below, the ‘dependency[i].link’ and the ‘dependency[i].depend’ means whether the slot *i* is subordinate to other slot, or the slot *i* is coordinate with other slot. When ‘dependency[i].depend’ is ‘1’, the slot *i* is subordinate to other slot, and when ‘dependency[i].depend’ is ‘0’, the slot *i* is coordinate with other slot.)

1. In starting point, neither comma nor ‘and’ is included.
  - 1.1 ‘C’: In case of that-clause, relative-clause and interrogative-clause, the clause is subordinate to the preceding clause.
    - dependency[i].link = just preceding slot index.
    - dependency[i].depend = 1.
  - 1.2 the others: The clause is coordinate with the preceding clause.
    - dependency[i].link = the starting point index of the preceding clause.
    - dependency[i].depend = 0.
2. In starting point, comma or ‘and’ is included.
  - 2.1 “(C) n V”, “and (C) n V”: The clause is coordinate with the most similar clause in ahead.
    - dependency[i].link = the starting point index of the most similar clause in ahead.
    - dependency[i].depend = 0.

The similarity between clauses is a weighted summation of the similarity between conjunctions, the subjects, and the verbs. The information referenced for these similarities is as follows.

- similarity between conjunctions: lexical similarity, type of conjunction
- similarity between the subjects: lexical similarity, type, semantic category of the headword of noun phrase
- similarity between the subjects: lexical similarity, type, tense, auxiliary verb of the verbs.

- 2.2 “, (n) V”: In case there is a starting point without “(n) V” in ahead, current starting point is linked to that starting point.

dependency[i].link = the slot index of starting point without “(n) V” in ahead  
 dependency[i].depend = 0;

This is the case there are included clause such as “Lack of water, as anybody who has a lawn knows, kills grass.”. The starting point of included clause is subordinate to the main clause.

- 2.3 “, and”, “, but” or “:”: In case “, and”, “, but” or “:” divide a clauses, the clause is not dependent on any clause in ahead, that is, the sentence is separated at this points.

dependency[i].link = -1,

```

dependency[i].depend = 0;
2.4 the others: The clause is coordinate with the
  just preceding clause.
dependency[i].link = just preceding slot index
dependency[i].depend = 0;

```

When the dependency analysis is done, main verbs are linked to their starting point. We can assume that basically each clause has its one main verb. So, if a starting point candidate has no main verb, that candidate must be removed from starting points. Also, we can resolve the range of a relative clause or noun clause included in a sentence by matching a main verb to a starting point its clause.

By linguistic constraints, the link between a starting point of a clause and the main verb of the clause must not be crossed. So, for linking a main verb of a clause to the starting point of the clause, whenever we meet a main verb, we link the verb to the right-most starting point which isn't linked to its main verb in the left-side starting points. Figure 2 shows such example evidently. In the figure, the ranges of what-clause and that-clause also are determined. The what-clause ends at the previous slot of the main verb of that-clause, and the that-clause ends at the previous slot of the main verb of whole sentence.

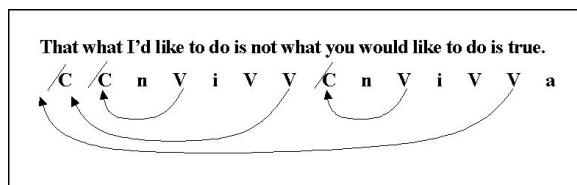


Figure 2. Linking main verbs to starting point of its clause.

The method of determining the dependency of a main verb which is  $i$ -th slot is as follows.

- $dependency[i].link$  = the slot index of the right-most starting point which isn't linked to its main verb in the left-side starting points.

- $dependency[i].depend = 0$ ;

The following is an example of the dependency analysis of the sentence in section 4.1, where 'x' represents no link.

#### [Result of Dependency Analysis]

index	link	depend
0 /n	-1	0
1 /C	0	1
2 V	1	0
3 p	x	x
4 V	0	0

5 /C	4	1
6 n	x	x
7 V	5	0
8 n	x	x
9 C	x	x
10 n	x	x
11 /C	10	1
12 V	11	0
13 p	x	x

#### 4.4 Determination of Depth

A depth of a clause is related with the depth in syntactic tree. In other words, as the depth is larger, the parsing of that-clause must be conducted earlier. By the depth of clauses, we can determine the ranges of sub-sentences and the sequence of translation of clauses. A depth of a slot can be obtained based on dependency list. If the dependency link of the slot exists, the depth of slot  $i$  is the summation of the depth of the slot which the link points to and  $dependency[i].depend$ . The algorithm of determining the depth of slot  $i$  is as follows.

```

depth[-1] = 0;
if exist dependency[i].link
  depth[i]=depth[dependency[i].link]+
  dependency[i].depend;

```

```

else
  depth[i]=depth[i-1];

```

(In above, index '-1' is the index of virtual slot indicating the beginning of a sentence.)

The following is the example about the sentence in section 4.1

#### [Calculation of Depth]

index	link	depend	depth
-1			0
0 /n	-1	0	$depth[-1]+0 = 0$
1 /C	0	1	$depth[0]+1 = 1$
2 V	1	0	$depth[1]+0 = 1$
3 p	x	x	$depth[2] = 1$
4 V	0	0	$depth[0]+0 = 0$
5 /C	4	1	$depth[-1]+0 = 1$
6 n	x	x	$depth[5] = 1$
7 V	5	0	$depth[5]+0 = 1$
8 n	x	x	$depth[7] = 1$
9 C	x	x	$depth[8] = 1$
10 n	x	x	$depth[9] = 1$
11 /C	10	1	$depth[10]+1 = 2$
12 V	11	0	$depth[11]+0 = 2$
13 p	x	x	$depth[12] = 2$

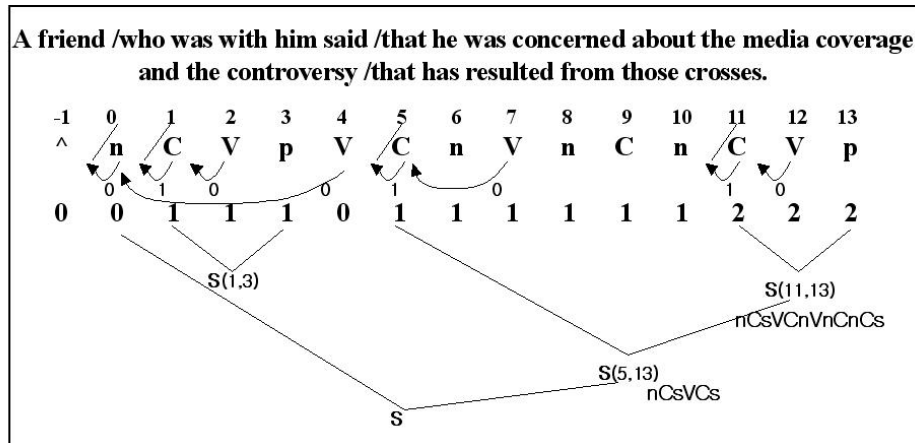


Figure 3. Partitioning by the depth

## 5 Sentence Partitioning

For the partitioning, we first detect the range of simple sentence, and then the range of wider clause by stages. It is achieved by recognizing the range according to the sequence that the depth is higher.

The method of recognizing only simple sentence has some advantages. Because the range of the simple sentence is not affected by the dependency of clauses, it is safe though there are some errors in clausal dependency analysis. Also, it is easy to describe a pattern for translating sentence pattern in which simple sentence reduced. In a sentence, simple sentence is defined as a clause which doesn't have that-clause or relative-clause for next clause. But, the method of the recognizing only simple sentence has a problem that it cannot reduce the length of sentence enough in case the input sentence is extraordinarily long. So, additional recognition of a sub-sentence with wider range is necessary, and it can be achieved by using depth.

Figure 3 shows such stages. In the figure, from the first number line, each line means slot index, slot pattern, dependency link and relation and the depth. The ranges of simple sentence are (1,3), (11,13), which represent pairs of the starting slot index and ending slot index of simple sentences. The sentence pattern in which simple sentences are reduced is 'nCsVnVnCs'. And the range of the clause with 2-depth range is (5,13). Then the sentence pattern results in 'nCsVCs'. Like this, the sub-sentence ranges are obtained by detecting the ranges with locally maximum depth.

## 6 Experimental Results

We use 108 sentences extracted from CNN news scripts as test sentences. Only 30 word-long sentences are selected. The average number of clause of the test sentences is 3.04, and the average depth of the test sentences is 0.7. The test sentences are analyzed by our method and compared to correct answers analyzed manually.

In the experiment, the clausal structure analysis is performed. And the sub-sentence range is extracted according to the depth by stages. For the first step, the ranges of simple sentences, which have the deepest depth, is extracted. The precision of recognizing all simple sentences is 85.2%. For the next step, the ranges of sub-sentence which cover 2-depth ranges are extracted in the clauses with a deeper depth than 0, and the precision is 79.3%. As predicted, the precision of the ranges of a wider clause is lower than that of narrower one.

The Figure 4 shows the partitioning points and the depth of each slot about the input sentence "The school has been secured and there is little else that anyone can do now to help the situation, except to stay away and let the sheriff's office contact you.". From the result, simple sentence ranges such as (0,1), (6,10), (11,15) are extracted and 2-depth range such as (2,10) is extracted.

The errors are mainly caused by parallel noun phrases, included clauses, and verb phrases connected by the conjunction 'and'. And there are also an ambiguity such as whether the word 'that' is used as sub-ordinate conjunction, or as relative pronoun, and whether the word 'who' is used as a relative, or as a interrogative. It should be noticed that the experiment is performed on

the assumption that the result of the tagging and partial parsing is correct. So, if we consider the tagging and partial parsing, the accuracy of whole processing will be considerably lower.

There is some difficulty in the quantitative comparison with other sentence partitioning methods, because the goal and the final result is somewhat different, but for the present, we can point out the fact that we use more information including the DB pattern for the ellipsis of a conjunction than other methods.

<input checked="" type="checkbox"/>	00, n &_so_COMMA right_now the moore school_system
<input type="checkbox"/>	01, V be close
<input checked="" type="checkbox"/>	02, C and[coor]
<input type="checkbox"/>	03, C as_soon_as
<input type="checkbox"/>	04, n we
<input type="checkbox"/>	05, V find_out
<input checked="" type="checkbox"/>	16, C when+
<input type="checkbox"/>	17, n they
<input type="checkbox"/>	18, V <be+._going_to>_#1:VERB
<input type="checkbox"/>	19, V reopen up
<input type="checkbox"/>	110, T COMMA
<input checked="" type="checkbox"/>	011, n we
<input type="checkbox"/>	012, V will let
<input type="checkbox"/>	013, n the public
<input type="checkbox"/>	014, V know
<input type="checkbox"/>	015, C when

Figure 4. A result of partitioning

## 7 Conclusion

For more efficient and high-quality translation of a long sentence, we develop a method of partitioning the long sentence. Our method analyzes the clausal structure effectively by the whole sentence pattern information. Then, adequate sub-sentence ranges are extracted from by stages. Experimentally, our partitioning of simple sentence shows 85.2% accuracy. With the partitioning, we can reduce the length and the syntactic ambiguity of sentence to translate. So, we can enhance the quality and coverage of translation.

Future works are as follow: (1) We should develop more systematic method for analyzing the clausal coordination and noun phrase coordination. (2) The describing method of the DB pattern for the ellipsis of conjunction need to be enhanced in the feature description. (3) The objective criterion evaluating our method should be developed.

## References

- [Choi et al., 1999] Sung-Kwon Choi, Taewan Kim, Sang-Hwa Yuh, Han-min Jung, Chul-Min Sim, Sang-kyu Park(1999) "English-to-Korean Web Translator: "FromTo/Web-EK"", *MTSUMMIT99*, Singapore.
- [Kim and Ehara, 1994] Yeun-Bae Kim, Terimasa Ehara, 1994, A Method for Partitioning of Long Japanese Sentences with Subject Resolution in J/E Machine Translation, *Proceedings of the 1994 ICCPOL*, May 10-13, Taejon, Korea.
- [Kim and Kim, 1997] Sung Dong Kim and Yung Taek Kim, 1997, Sentence Sengmentation for Efficient English Syntactic Analysis, *Journal of KISS(B)* Vol.24 No.8 (Korean).
- [Kurohashi and Nagao, 1994] Sadao Kurohashi and Makoto Nagao, 1994, A Syntactic Analysis Method of Long Japanese Sentences Based on the Detection of Conjunctive Structures, *Computational Linguistics* Vol.20 No.4, pp507-534
- [Okumura and Muraki, 1990] Akitoshi Okumura and Kazunori Muraki, 1990, Symmetric Pattern Matching Analysis for English Coordinate Structures. In *Proceeding of 4<sup>th</sup> Conference on Applied NLP*.
- [Peh and Ting, 1996] L. S. Peh and Christopher H. A. 1996. A Divide-and-Conquer Strategy for Parsing. In *Proceedings of the ACL/SIGPARSE 5<sup>th</sup> International Workshop on Parsing Technologies*.
- [Seo et al., 2001] 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, *MTSUMMIT 2001*.
- [Yoon and Song, 1997] Juntae Yoon and Mansuk Song, 1997, Analysis of Coordinate Conjunctive Phrase In Korean, *Journal of KISS(B)* Vol.24 No.3 (Korean).