Syntactic Category Prediction for Improving Translation Quality in English-Korean Machine Translation*

Sung-Dong Kim

Department of Computer Engineering, Hansung University,
Seoul, 136-792, Republic of Korea
sdkim@hansung.ac.kr

Abstract. This paper proposes the syntactic category prediction for improving translation quality. In parsing using sentence segmentation, the segments are separately parsed and then the parsing results of each segment are combined to generate a global sentence structure. The syntactic category prediction guides the parser to identify relationships among segments and to select the correct parsing results for each segment. We design features for predicting syntactic categories and generate decision trees for the prediction using training data from the Penn Treebank. In experiment, we show the prediction accuracy and comparison results with the prediction by human-built rules, heuristic probability function, and neural networks. Also, we present how much the category prediction contributes to improving translation quality.

Keywords: machine translation, machine learning, syntactic category prediction.

1 Introduction

Recent English-Korean machine translation systems generate good translation for the relatively short sentences. In the translation of long sentences, the translation results are bad, so the readers have difficulty in understanding the meaning of translated sentences. The difficulty in translating long sentences is syntactic one, while the problems in short sentence translation lie in the semantic area. That is, more accurate parsing helps improve the readability of the translation results for long sentences. Most long sentences consist of comma-separated phrases or clauses. The accurate and detailed analysis of the relationships among the comma-separated elements can improve the parsing accuracy, resulting in translation quality improvement. Of course, the semantic problems must be considered to improve the translation quality, but they are not scope of the paper.

In (Kim, 2005; Kim et al., 2001), they proposed intra-sentence segmentation for speeding up the syntactic analysis of long sentences. In parsing using the segmentation, the input sentece is split into several shorter segments by commas and the above intra-sentence segmentation. The segments are parsed separately and the parsing results of segments are combined. After parsing each segment, a tree must be selected. So several selection decisions occur during parsing an input sentence. The wrong selection affects the translation result. As a result, the intra-sentence segmentation contributed to speeding up the parsing but may make little improvement of translation quality. Also, it is difficult to consider the long-distance dependencies amone segments, which can lead to additional translation errors.

In order to improve translation quality by considering long-distance dependencies and selecting the correct parsing results for each segment, this paper proposes a syntactic category prediction of comma-separated segments. If we could know the syntactic category of a given

* This research was financially supported by Hansung University in the year of 2009.

segment before parsing, we can guide parser in considering segment dependencies and selecting the correct parsing results. The prediction must be made before parsing using only information from lexical analysis. A sentence is split by commas, and then the long segments are again split. We try to predict the syntactic category of the comma-separated segments. This prevents predicting categories of the non-constituent phrases from the second segmentation step. In this paper, we construct rules and functions for the syntactic category prediction by the statistical and machine learning methods. We generate training data using the Penn Treebank corpus (Marcus et al., 1993).

Section 2 describes the parsing method using sentence segmentation and syntactic category prediction. We explain the generation steps of rules and functions for the syntactic category prediction in section 3. Section 4 shows the comparison results of prediction accuracies and the degree of the translation quality improvement. Section 5 concludes the paper.

2 Related Works & Parsing Method

The partial parsing by Abney (Abney, 1991) was used in analyzing noun phrases and prepositional phrases and was regarded as an origin for fast parsing. In (Kim and Kim, 1995), they proposed sentence patterns in parsing English sentences. The method was effective for the sentences matched with the defined patterns. However, the coverage of the sentence patterns was very low for practical usage. In (Kim et al., 2001; Kim and Kim, 1997), the intra-sentence segmentation and the method of partial parsing were used to improve parsing efficiency.

In general, long sentences consist of comma-separated segments. The segments have some roles in a sentence which are determined by the syntactic category of each segment. The predicted category can be considered in identifying relationships among segments and can help select a correct parsing result for each segment.

There are some works on the category prediction. Most works are for the word category prediction in the spoken language analysis. They used neural network (Nakamura, 1995), n-grams, and so on. Others works referred the category prediction as part-of-speech tagging. There are many researches about POS tagging. They use HMM, neural networks, n-grams, and probabilistic methods (Nivre, 2000; Schmid, 1994). But this paper focuses on the prediction of the syntactic categories of comma-separated segments. In the literature, there are few researches about the problem of syntactic category prediction.

A small TCL interpreter, which can be linked into the code, interprets the strings.

Segment-1: A small TCL interpreter → **NP**
Segment-2: which can be linked into the code → **RLCL**
Segment-3: interprets the strings → **VP**

Translation-1: 작은(small) TCL 인터프리터(interpreter), 그 코드(code)로 연결(link)될 수 있는 것(which), 그 문자열(strings)을 해석한다(interprets).
Translation-2: 그 코드(code)로 연결(link)될 수 있는 작은(small) TCL 인터프리터(interpreter)는 그 문자열(strings)을 해석한다(interprets).

Figure 1: Examples of segments, their categories, and possible translations.

Figure 1 shows the examples of comma-separated segments, the categories of the segments, and possible translations. In Figure 1, [Translation-1] is a translation result without category prediction. In the translation NP (noun phrase) result is selected for [Segment-1], SENT result is for [Segment-2], and VP result is for [Segment-3]. The selection for [Segment-2] is wrong, but the error can be easily solved by modifying some scores in the scoring system. But this solution interferes with the scoring system. We can get correct translation as [Translation-2]

when we identify modification relation between [Segment-1] and [Segment-2] and choose RLCL result for [Segment-2] by predicting its category as RLCL.

The syntactic category prediction helps to identify relationship between segments and select a correct parsing result for a given segment without affecting existing parser. Figure 2 shows the parsing steps with the syntactic category prediction. After segmentation by commas, syntactic category prediction is performed on each segment. There are exceptions in which the commas are not used to separate phrases. For example, in "a very heavy, expensive book", neither "a very heavy" or "expensive book" is a linguistic phrase. We use "comma rewriting" step in pre-processing steps before lexical analysis. By the step, the phrase is rewritten to "a very heavy and expensive book." This "comma rewriting" is done automatically using the comma rewriting rules. For parsing a sentence, we build a parsing control tree in which a node is for a comma-separated segment. At first, all nodes are connected as sibling relation. With the knowledge about the segment relationships, a node for modifier segment becomes child node of a node for modified segment and one node in the tree is selected as root. A segment included in a node is parsed separately, but parsing result of a segment in a child node participates in parsing a segment of a parent node. When a segment included in the root node is parsed, all parsing results of other segments are used to build parsing trees for the whole sentence. This is the work done in the "synthesis of results" step in Figure 2. Then, a resulting parse tree is passed to Korean generation module (Yang, 1997).

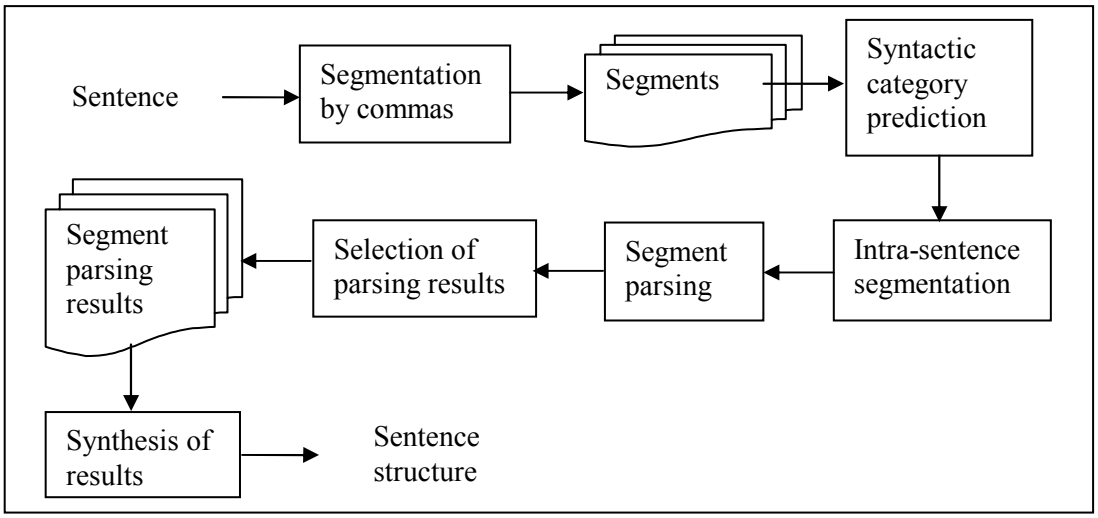

Figure 2: Parsing steps with the syntactic category prediction.

3 Methods of Syntactic Category Prediction

This section explains the generation of syntactic category prediction rules and functions using the Penn Treebank corpus. We adopt a heuristic probability function, decision trees and neural networks. Figure 3 shows the generation process of the rules and the functions.

3.1 Target Syntactic Categories

We choose 7 categories as the prediction target: SUBCL, RLCL, NP, VP, PP, AJP, AVP. The above choice is based on the experience during translation test for English-Korean machine translation system, SmarTran.

We exclude SENT because it is difficult to find features for predicting SENT category and the prediction as SENT gives little advantages in identifying relationships with other segments. For above 7 target categories, we expect the usefulness of the category prediction in parsing.

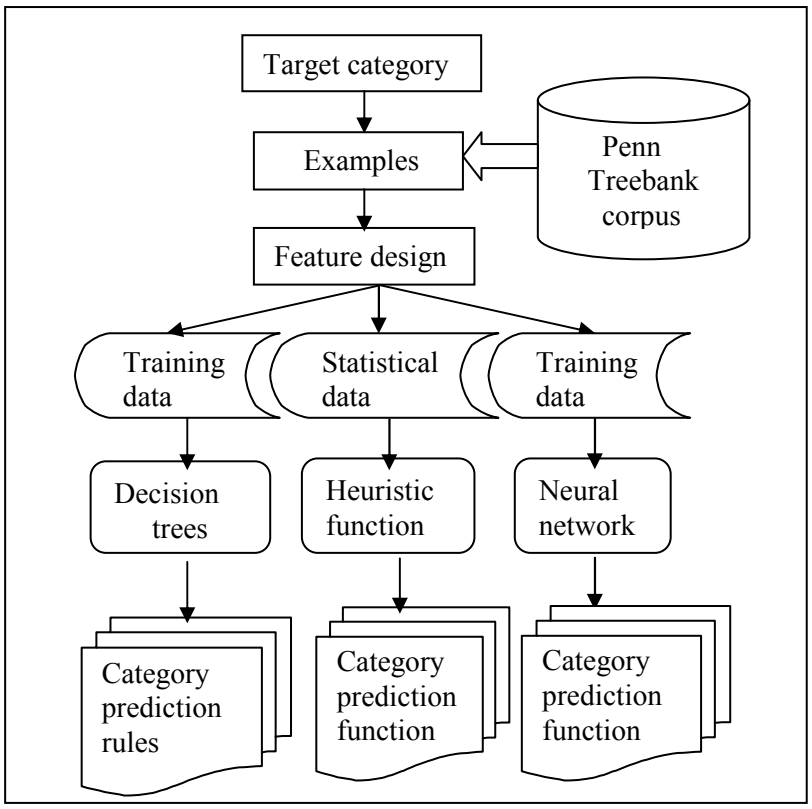

Figure 3: Generation process of the category prediction rules and functions.

3.2 Collecting Examples for Training Data

In this step, we collect phrases and clauses for the target categories from the Penn Treebank corpus. In processing the corpus for collecting examples for syntactic categories, we separate parsed results by commas and represent the separated result (segment) as a series of "word/POS tag" and its category. Figure 4 shows some collected examples.

```
( (S
  (NP-SBJ  (NP (NNP Pierre) (NNP Vinken) )  (, ,)
  (ADJP  (NP (CD 61) (NNS years) ) (JJ old) ) (, ,) )
  (VP (MD will)  (VP (VP join) (NP (DT the) (NN board) )
                   (PP-CLR (IN as)  (NP (DT a) (JJ nonexecutive) (NN director) ))
                           (NP-TMP (NNP Nov.) (CD 29) )))
  (. .) ))
```

(a)

Pierre/NNP Vinken/NNP **NP**
61/CD years/NNS old/JJ **AJP**
will/MD join/VB the/DT board/NN as/IN a/DT nonexecutive/JJ director/NN Nov./NNP 29/CD **VP**

(b)

Figure 4: Collected examples for training data generation.

In Figure 4, (a) is the parsing result for "*Pierre Vinken, 61 years old, will join the board s a nonexecutive director Nov. 29.*" This is an example of the Penn Treebank corpus from which

we generate examples as in (b). There are three segments separated by commas as shown in (b). In (b), the last bold font words (**NP**, **AJP**, **VP**) are the category of their segment. They are examples from which we construct training data for the generation of the prediction rules and functions.

3.3 Feature Design

We use 7 features which must be considered in determining syntactic category. They are first word/tag and the second word/tag of the segment, the last tag, the length of the segment, and the clue for the clause. The last feature, clue for the clause, represents whether some part-of-speeches exist in the segment, which are for the relative pronoun (WDT, WP) and can be a main verb of a clause (MD, VBD, VBZ, and VBP).

3.4 Training Data & Generation of Prediction Rules and Functions

This section describes training data and generation process of the prediction rules and functions. Figure 5 shows some examples of training data for decision trees and neural network. In Figure 5, (a) is the examples in Figure 4 from which we generate (b) and (c). The training data for decision tree learning is (b), and (c) is for neural network. We discretize the word features to the numerical values.

Pierre/NNP Vinken/NNP **NP**
61/CD years/NNS old/JJ **AJP**
will/MD join/VB the/DT board/NN as/IN a/DT nonexecutive/JJ director/NN Nov./NNP 29/CD **VP**

(a)

2, NNP, NNP, *NONE*, 2, NNP, 2, NP
0, CD, JJ, *NONE*, 4139, NNS, 3, AJP
4085, MD, CD, MD, 1920, VB, 10, VP

(b)

2	13	13	35	2	13	2	0
0	1	6	35	4139	12	3	3
4085	10	1	10	1920	25	10	1

(c)

Figure 5: Examples of training data.

Of 22,112 training data, we use 3,000 data for validation and the remains for neural network training. We adopt early stopping to terminate the training at the point where the validation error increases. The scaled conjugate gradient (Moller, 1993) is used for the training algorithm. The neural network consists of two hidden layers, where we use *tansig* (hyper tangent) function, and output layer, where we use *sig* (sigmoid) function. After using various neural network structures, we select the model which shows the least validation error. The resulting neural network has 15 nodes at first hidden layer, 10 nodes at second hidden layer, and 7 target neurons.

We use a simple probability function based on Bayes' rule as a baseline for category prediction. Given a segment, we consider the first tag, the second and the last tags. We calculate the probability of a segment category *C*, given above three tags which are from the segment. Thus, the most probable syntactic category *C'* is determined as follows:

$$C' = \arg\max_C \Pr(C \mid \text{three tags}) \quad = \arg\max_C \frac{\Pr(C)\Pr(\text{three tags} \mid C)}{\Pr(\text{three tags})} \tag{1}$$

In equation (1), the denominator part is same for a given segment. Therefore, we can express the syntactic category prediction as equation (2).

$$C' = \arg\max_{C} \Pr(C) \Pr(\text{three tags} \mid C) \quad (2)$$

4 Experiment

4.1 Data

We generate the training and the test data from the Penn Treebank corpus. In collecting training data described in section 3.4, the parsing results of 19,697 sentences from Wall Street Journal are used. Table 1 shows the distribution of the training data according to the syntactic categories. In the table, "# of data" means the number of comma-separated segments. For category prediction test, we extract data from Wall Street Journal, Brown corpus, IBM manual, and ECTB (English Chinese Tree Bank) corpus. We use several domains to get test data from much different domains as possible. This is why we try to show that the proposed method will be domain-independent. Table 2 shows the distribution of the test data. We construct another version of data for measuring how much the category prediction would contribute to improving translation quality. We extract 100 sentences from computer and politics domains, Wall Street Journal, and high school English text book, resulting in 400 test sentences.

Table 1: Distribution of the training data.

Target category	# of data
NP	14,091
VP	1,499
AJP	3,806
AVP	334
PP	1,585
SUBCL	751
RLCL	46
Total	22,112

Table 2: Test data for prediction accuracy evaluation.

Domain	# of sentences	# of data
WSJ	4,000	4,626
Brown	4,001	3,821
IBM	4,404	1,403
ECTB	3,825	4,834
Total	16,230	14,684

4.2 Evaluation of Category Prediction Methods

For performance comparison, we use 4 methods for category prediction. Three of them are described in section 3.4 (decision trees, neural networks, a heuristic probability function). In addition, we use a prediction method by human-built rules which were collected during development of English-Korean MT system, a test bed of the research. The accuracy is defined as the number of correct prediction over all comma-separated segments in test data. Table 3 shows the comparison result. The best accuracy in each domain is indicated by bold font digits. The prediction rules generated by the decision tree learning (DT) shows the best prediction accuracy in all domains.

Table 3: Prediction accuracy (%).

	DT	Prob. function	Human-built rules	Neural network
WSJ	**97**	92	88	95
Brown	**94**	88	83	91
IBM	**97**	78	85	86
ECTB	**97**	90	90	96
Average	**96**	89.1	87.1	93.4

Figure 6 shows some examples the prediction rules converted from the decision tree results. The rules read off from the decision trees have the advantage that they are easy to interpret and to be improved by an additional error correction process, while it is difficult to interpret the prediction process by the probability function or neural network methods.

```
if (!strcmp(szFirstWord,"that") && check_IN(firstTrees)){
  setTargetCategory( THAT_CLAUSE ); return;
}
if (!strcmp(szFirstWord,"although") && check_IN(firstTrees)) {
  setTargetCategory( SUBCL );  return;
}
if (check_IN(firstTrees) && checkClauseExist(_NONE) &&
  check_NNP(secondTrees)) {
  setTargetCategory( PP );  return;
}
if (check_CD(firstTrees) && check_NONE(lastTrees)) {
  setTargetCategory( NP );  return;
}
```

Figure 6: Examples of prediction rules from decision tree learning.

4.3 Evaluation of Translation Quality Improvement

We integrate 155 prediction rules read off from the decision trees with the English-Korean MT system to evaluate the translation quality improvement by the syntactic category prediction. The EKMT system performs rule-based analysis and adopts idiom translation approach.

Figure 2 in section 2 shows the parsing steps in our MT system. There is a prediction process before "segment parsing", "selection of parsing results" and "synthesis of results." Thus, a predicted category can be used to guide the segment parsing and the selection of segment parsing result. Also the identified relationships among segments by the predicted category can help "synthesis of result". By this, we expect the translation quality improvement.

Table 4 shows the results of translation evaluation by seven people. They compare two translation results by marking better/equal/worse. In the table, "Equal" means that the evaluator thinks the two translation results have the same meaning or it is difficult to identify the superior translation. The results indicate the fact that the introduction of syntactic category prediction contributed to generating more the better translations than the worse translations. In order to speed up translation it makes sense to split long sentences into segments and translate them one-by-one. However, this neglects the long-distance dependencies, which can lead to additional translation errors. If the parsing algorithm considers the long-distance dependencies among split segments by using category prediction results, it is possible to recover from those errors. As a result, we conclude that the syntactic category prediction can play a role for the translation quality improvement.

Table 4: Translation quality improvement.

	Better	Equal	Worse
WSJ	56	18	26
Computer	53	11	36
Politics	60	12	28
Text book	66	10	24
Total	235 (58.8%)	51 (12.7%)	114 (28.5%)

5 Conclusion

This paper proposes the syntactic category prediction to improve the translation quality in English-Korean machine translation. We construct the prediction rules and functions using statistical and machine learning methods. In parsing long sentences, the sentences are split into shorter segments by commas. The purpose of the syntactic category prediction is to improve the translation quality by identifying relationships among comma-separated segments and guiding the selection of the accurate analysis results of the segments.

We construct training data using the Penn Treebank corpus. We apply decision tree learning and the neural network learning for the generation of the prediction rules and the functions. The rules by the decision tree learning show the best prediction accuracy, so they are integrated into the English-Korean MT system. The predicted syntactic category is used to identify segment relationships and select the parse trees for the analyzed segments. The parser does not consider the predicted category when generating parse trees for the segments. The proposed syntactic category prediction is not for the general parser, but for the translation-adapted parser. That is, the syntactic category prediction aims to only improve the translation quality. Through the translation evaluation, we know that the prediction contributes to improving the translation quality.

In order to improve the prediction accuracy, approaches using other classifiers, such as SVM and maximum entropy, will be considered. The syntactic category prediction can be used to filter the parsing rules so that parsing efficiencies in time/space may be improved. We expect that the syntactic category prediction will contribute to designing the new approach to parsing of long sentences composed of comma-separated segments.

References

Abney, S. 1991. Parsing By Chunks. *Principle-Based Parsing*. Kluwer Academic Publishers. pp. 257-279.

Kim, S.-D. 2005. Intra-sentence Segmentation based on Maximum Entropy Model for Efficient English Syntactic Analysis. *Journal of Korean Information Science Society*, 32(5), 385-395.

Kim, S. D. and Y.T. Kim. 1995. Sentence Analysis using Pattern Matching in English-Korean Machine Translation. *In International Conference on Computer Processing on Oriental Languages*, pp. 199- 206.

Kim, S.-D. and Y.T. Kim. 1997. Intra-sentence Segmentation for Efficient English Syntactic Analysis. *Journal of Korean Information Science Society*, 24(8), 884-890.

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.

Marcus, M. P., B. Santorini and M.A. Marcinkiewicz. 1993. Building a Large Annotated Corpus of English: The Penn Treebank, *Computational Linguistics*, 19(2), 313-330.

Moller, M. F. 1993. A scaled conjugate gradient algorithm for fast supervised learning. *Neural Networks*, 6, 525-533.

Nakamura, M., K. Tsuda and J.-I. Aoe. 1995. Word category prediction based on neural network, *International journal of computer mathematics*, 57(3), 169-181.

Nivre, J. 2000. Sparse data and smoothing in statistical part-of-speech tagging. *Journal of Quantitative Linguistics*, 7(1), 1-18.

Schmid, H. 1994. Part-of-Speech Tagging with Neural Networks. *In 15th International Conference on Computational Linguistics*, 172-176.

Yang, S.H. 1997. Transfer of Linguistic Style for English-to-Korean Machine Translation. *Ph.D Dissertation of Seoul National University*.