

# Improving Translation Quality of Rule-based Machine Translation

Paisarn Charoenpornswat, Virach Somlertlamvanich and Thatsanee Charoenporn  
Information Research and Development Division  
National Electronics and Computer Technology Center  
112 Thailand Science Park, Paholyothin Rd.,  
Klong 1, Klong Luang, Pathumthani 12120  
THAILAND  
{paisarn, virach, thatsanee}@nectec.or.th

## Abstract

This paper proposes machine learning techniques, which help disambiguate word meaning. These methods focus on considering the relationship between a word and its surroundings, described as context information in the paper. Context information is produced from rule-based translation such as part-of-speech tags, semantic concept, case relations and so on. To automatically extract the context information, we apply machine learning algorithms which are C4.5, C4.5rule and RIPPER. In this paper, we test on *ParSit*, which is an interlingual-based machine translation for English to Thai. To evaluate our approach, an verb-to-be is selected because it has increased in frequency and it is quite difficult to be translated into Thai by using only linguistic rules. The result shows that the accuracy of C4.5, C4.5rule and RIPPER are 77.7%, 73.1% and 76.1% respectively whereas *ParSit* give accuracy only 48%.

## Introduction

Machine translation has been developed for many decades. Many approaches have been proposed such as rule-based, statistic-based [5], and example-based approaches [3, 6, 11]. However, there is no machine learning technique that meets human's requirement. Each technique has its own advantages and disadvantages. Statistic-based, example-based and corpus-based approaches were recently proposed. A rule-based approach is the first strategy pursued by research in the field of machine translation. Rules are written from linguistic knowledge by human. The strength is that it can deeply analyze in both syntax and semantic levels. However, the

weak points of this model are 1) it requires much linguistic knowledge. 2) it is impossible to write rules that cover all a language. In many years ago, a statistic-based and an example-based were proposed. These approaches do not require linguistic knowledge, but they need large size of bilingual corpus. A statistic-based approach uses statistic of bilingual corpus and language model. The advantage is that it may be able to produce suitable translations even if a given sentence is not similar to any sentences in a training corpus. In contrast, an example-based can produce appropriate translations in case of a given sentence must similar to any sentences in a training data. Nevertheless, a statistic-based approach cannot translate idioms and phrases that reflect long-distance dependency.

To improve quality of a rule-based machine translation, we have to modify/add some generation rules or analysis rules. This method requires much linguistic knowledge and we cannot guarantee that accuracy will be better. For example, in case of modifying some rules, it does not only change incorrect sentences to correct sentences furthermore they may effect on correct sentences too. The common errors of machine translation can be classified into two main groups. One is choosing incorrect meaning and the other is incorrect ordering. In our experiments, we select *ParSit* in evaluation. *ParSit* is English-to-Thai machine translation by using an interlingual-based approach [8]. An interlingual-based approach is a kind of rule-based machine translation. The statistics of incorrect meaning and incorrect ordering in *ParSit* are 81.74% and 18.26% respectively. Therefore, in this paper, we address on choosing a correct meaning. We use context information,

words and part-of-speech tags, in classifying the correct meaning. This paper, we apply machine learning algorithms, C4.5, C4.5rule, and RIPPER, to automatically extract words and part-of-speech tags.

### 1. A Rule-Based Approach: Case Study *ParSit*: English to Thai Machine Translation.

In this section, we will briefly describe a rule-based machine translation. Each rule-based machine translation has its own mythology in translation. Hence in this paper, we select *ParSit* as a case study. *ParSit* is English to Thai machine translation using an interlingual-based approach. *ParSit* consists of four modules that are a syntax analysis module, a semantic analysis module, a semantic generation module, and a syntax generation module. An example of *ParSit* translation is shown in figure 1.

In figure 1, the English sentence, “*We develop a computer system for sentence translation.*”, input into *ParSit*. Both syntax and semantic analysis modules analyze the sentence and then transform into the interlingual tree which is shown in Figure 1. In the interlingual tree shows the relationship between words such as 1) “*We*” is an agent of “*develop*” 2) “*system*” is an object of “*develop*” 3) “*computer*” is modifier of “*system*” and so on. Finally, Thai sentence, *พวกเราพัฒนาระบบคอมพิวเตอร์เพื่อการแปลประโยค*, is generated from the interlingual tree by the syntax and semantic generation modules.

The errors of translation from *ParSit* can be classified into two main groups. One is incorrect meaning and the other is incorrect ordering. The incorrect meaning also can be reclassified into three categories; 1). missing some words 2). generating over words 3). using incorrect word The examples of errors are shown below.

- *Incorrect meaning errors.*
  - Missing some words.
 

The city is not far from here  
 Incorrect: เมืองไม่ไกลจากนี้  
 Correct: เมืองอยู่ไม่ไกลจากนี้

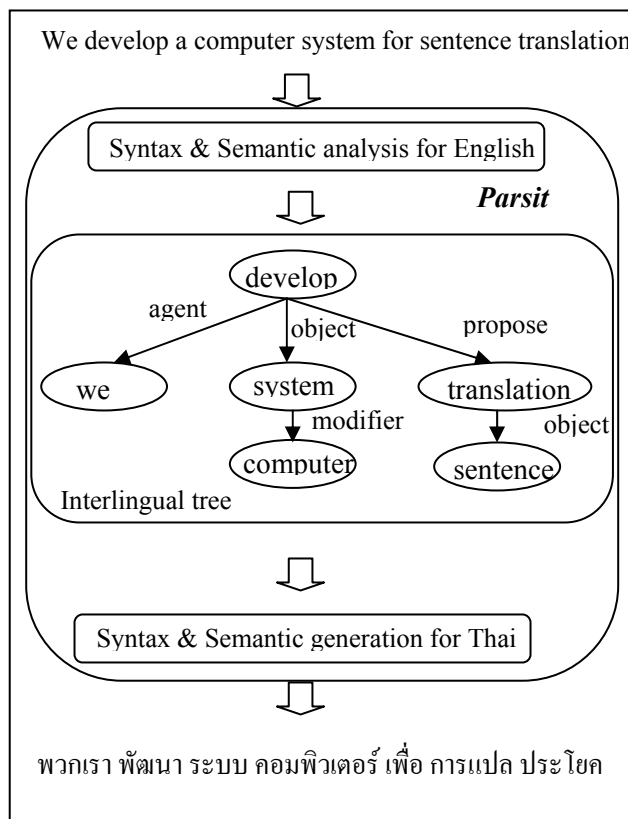


Figure 1: *ParSit* translation process.

- Generating over words.
 

This is the house in which she lives.  
 Incorrect: นี่คือนบ้านที่เธออาศัยอยู่ที่นั่น  
 Correct: นี่คือนบ้านที่เธออาศัยอยู่
- Using an incorrect word.
 

The news that she died was a great shock.  
 Incorrect: ข่าวที่ว่าที่เธอตายเป็นที่ตกใจอย่างมาก  
 Correct: ข่าวที่ว่าที่เธอตายเป็นที่ตกใจที่ยิ่งใหญ่

- *Incorrect ordering errors.*

He is wrong to leave.  
 Incorrect: เขาจากไปผิดที่  
 Correct: เขาผิดที่จากไป

We evaluated *ParSit* by using 770-English-sentence corpus that is designed by Japan Electronic Industry Development Association (JEIDA). This corpus has the characteristics for testing in word level such as concept mismatching, word absence and etc. and sentence level such as grammar and modifier misplacement. The statistics of *ParSit* errors are shown in Table 1.

Table 1. Statistics of *ParSit* Error

| Incorrect meaning errors |       |       | Incorrect ordering errors (%) |
|--------------------------|-------|-------|-------------------------------|
| M (%)                    | G (%) | U (%) |                               |
| 16.71                    | 13.31 | 51.42 | 18.26                         |

In table 1, *M*, *G* and *U* mean missing some word errors, generating over word errors and using incorrect word errors respectively.

According to Table 1, *ParSit* makes many errors in choosing incorrect meaning (81.74%). In this paper, we focus on solving the problem of choosing incorrect meaning. To decide what is the correct meaning of a word, we propose to use context information around that word. Context information that we use will be described in the next section.

## 2 Applying Machine Learning Technique

### 2.1 Context Information

There are many kinds of context information that useful to decide the appropriate meaning of a word such as grammatical rules, collocation words, context words, semantic concept and etc. Context information is derived from a rule-based machine translation. Words and their part-of-speech tags are the simplest information, which are produced from English analysis module. In this paper, we use words and/or part-of-speech tags around a target word in deciding a word meaning.

### 2.2 Machine Learning

In this section, we will briefly describe three machine learning techniques, C4.5, C4.5rule and RIPPER.

#### 2.2.1 C4.5 & C4.5Rule

C4.5, decision tree, is a traditional classifying technique that proposed by Quinlan [7]. C4.5

have been successfully applied in many NLP problems such as word extraction [9] and sentence boundary disambiguation [2]. So in this paper, we employ C4.5 in our experiments.

The induction algorithm proceeds by evaluation content of series of attributes and iteratively building a tree from the attribute values with the leaves of the decision tree being the value of the goal attribute. At each step of learning procedure, the evolving tree is branched on the attribute that partitions the data items with the highest information gain. Branches will be added until all items in the training set are classified. To reduce the effect of overfitting, C4.5 prunes the entire decision tree constructed. It recursively examines each subtree to determine whether replacing it with a leaf or branch would reduce expected error rate. This pruning makes the decision tree better in dealing with the data different from training data.

In C4.5 version 8, it provides the other technique, which is extended from C4.5 called C4.5rule. C4.5rule extracts production rules from an unpruned decision tree produced by C4.5, and then improves process by greedily deletes or adds single rules in an effort to reduce description length. So in this paper we also employ both techniques of C4.5 and C4.5rule.

#### 2.2.2 RIPPER

RIPPER [10] is the one of the famous machine learning techniques applying in NLP problems [4], which was proposed by William W. Cohen. On his experiment [10] shows that RIPPER is more efficient than C4.5 on noisy data and it scales nearly linearly with the number of examples in a dataset. So we decide to choose RIPPER in evaluating and comparing results with C4.5 and C4.5rule.

RIPPER is a propositional rule learning algorithm that constructs a ruleset which classifies the training data [11]. A rule in the constructed ruleset is represented in the form of a conjunction of conditions:

if  $T_1$  and  $T_2$  and ...  $T_n$  then class  $C_x$ .

$T_1$  and  $T_2$  and ...  $T_n$  is called the body of the rule.  $C_x$  is a target class to be learned; it can be a positive or negative class. A condition  $T_i$  tests for a particular value of an attribute, and it takes one of four forms:  $A_n = v$ ,  $A_c \geq \theta$ ,  $A_c \leq \theta$  and  $v$

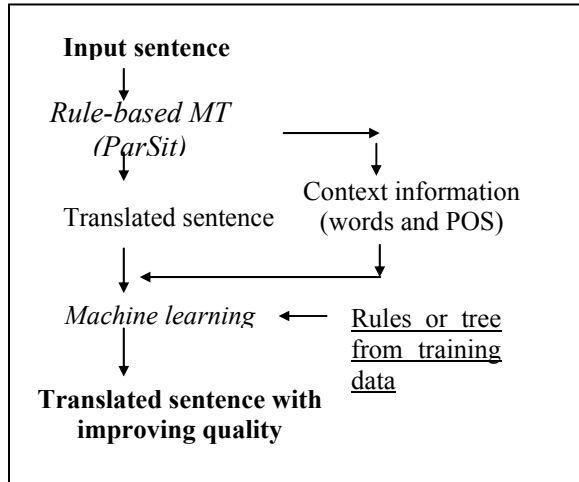


Figure 2 : Overview of the system

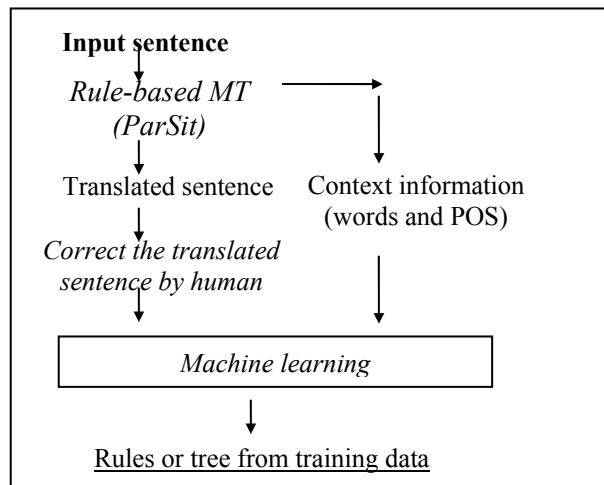


Figure 3 : The training module

$\in A_s$ , where  $A_n$  is a nominal attribute and  $v$  is a legal value for  $A_n$ ; or  $A_c$  is a continuous variable and  $\theta$  is some value for  $A_c$  that occurs in the training data; or  $A_s$  is a set-value attribute and  $v$  is a value that is an element of  $A_s$ . In fact, a condition can include negation. A set-valued attribute is an attribute whose value is a set of strings. The primitive tests on a set-valued attribute  $A_s$  are of the form “ $v \in A_s$ ”. When constructing a rule, RIPPER finds the test that maximizes *information gain* for a set of examples  $S$  efficiently, making only a single pass over  $S$  for each attribute. All symbols  $v$ , that appear as elements of attribute  $A$  for some training examples, are considered by RIPPER.

### 3 Overview of The System

In this section, we will describe the process of our system in Figure 2. First, input a source sentence into rule-based MT and then use syntax and semantic rules for analysing the sentence. At this step, rule-based MT gives various kinds of word information. In this experiment we used only words and part-of-speech tags. After analysing, rule-based MT generates a sentence into target language. Next, the translated sentence from rule-based MT and the context information are parsed into machine learning. Machine learning requires a rule set or a decision tree, which are generated from a training set, to decide what is the appropriate meaning of a word.

In training module (Figure 3), we parse English sentences with part-of-speech tags, which are given by *ParSit*, and assign the correct meaning by linguists into machine learning module. The machine learning will learn and produces a rule set or a decision tree for disambiguating word meaning. The process of training is shown in Figure 3.

### 4 Preliminary Experiments & Results.

To evaluate our approach, we should test on a word, which frequently occurred in normal text and has several meanings. According to the statistics of word usage from 100M-word British National Corpus, verb-to-be occurred more than three million times, and translation of verb-to-be into Thai is quite difficult by using only linguistic rules. Therefore our experiment, we test our approach on verb-to-be.

In the experiment, we use 3,200 English sentences from Japan Electronic Dictionary Research Institute (EDR). EDR corpus is collected from news, novel and journal. Then our linguists manually assigned the suitable meaning of verb-to-be in Thai. In training and testing steps, we divided data into two groups. The first is 700 sentences for testing and the other is for training. We use various sizes of a training data set and different sizes of context information.

Table 2, 3 and 4 are the result from C4.5, C4.5rule and RIPPER respectively. The series in columns represent the number of

training sentences. The row headers show the types of context information that Pos±n, Word±n and P&W±n mean part-of-speech tags, words and part-of-speech tags and words with the window size is n.

Table 2. The results from C4.5

|        | 100         | 500         | 1 K         | 1.5K        | 2K          | 2.5K        |
|--------|-------------|-------------|-------------|-------------|-------------|-------------|
| Pos±1  | <b>67.1</b> | <b>69.8</b> | 69.8        | 69.8        | 69.8        | 69.8        |
| Pos±2  | <b>67.1</b> | <b>69.8</b> | 69.8        | 69.8        | 69.8        | 69.8        |
| Pos±3  | <b>67.1</b> | <b>69.8</b> | 69.8        | 69.8        | 69.8        | 69.8        |
| Word±1 | 55.5        | 63.2        | <b>73.1</b> | <b>74.2</b> | <b>75.5</b> | 75.4        |
| Word±2 | 57.7        | 64.6        | 71.7        | 72.7        | <b>75.5</b> | 77.3        |
| Word±3 | 57.8        | 65.3        | 71.3        | 73.1        | 75.4        | <b>77.7</b> |
| P&W±1  | 55.5        | 68.6        | 71.1        | 71.3        | 71.8        | 71.8        |
| P&W±2  | 57.7        | 68.6        | 71.3        | 70.4        | 71.8        | 71.8        |
| P&W±3  | 57.8        | 68.6        | 71.3        | 69.6        | 71.3        | 71.9        |

Table 3: The results from C4.5rule

|        | 100         | 500         | 1 K         | 1.5K        | 2K          | 2.5K        |
|--------|-------------|-------------|-------------|-------------|-------------|-------------|
| Pos±1  | <b>69.8</b> | 71.3        | <b>76.3</b> | <b>77.3</b> | <u>76.0</u> | <b>73.1</b> |
| Pos±2  | <b>69.8</b> | <b>77.5</b> | <b>76.7</b> | <u>76.9</u> | <u>76.3</u> | <b>73.1</b> |
| Pos±3  | <b>69.2</b> | <u>77.2</u> | <b>76.2</b> | <u>76.8</u> | 70.1        | <b>73.1</b> |
| Word±1 | 54.9        | 73.1        | 63.4        | 63.6        | 67.2        | 71.1        |
| Word±2 | 56.3        | 73.5        | 73.5        | 72.5        | 64.7        | 70.6        |
| Word±3 | 56.3        | 72.2        | 72.5        | 72.3        | <b>76.8</b> | 70.6        |
| P&W±1  | 54.9        | <u>77.2</u> | 63.4        | 68.4        | 69.2        | 71.1        |
| P&W±2  | 56.8        | <u>76.7</u> | 73.5        | 68.0        | 70.5        | 70.6        |
| P&W±3  | 56.8        | 69.6        | 64.3        | 61.8        | 71.5        | 71.1        |

Table 4: The results from RIPPER.

|        | 100         | 500         | 1 K         | 1.5K        | 2K          | 2.5K        |
|--------|-------------|-------------|-------------|-------------|-------------|-------------|
| Pos±1  | 70.2        | 70.9        | <b>73.3</b> | 71.7        | 72.1        | <b>76.1</b> |
| Pos±2  | 69.4        | 71.0        | 69.2        | 70.2        | 70.8        | 72.1        |
| Pos±3  | 69.2        | 71.0        | 69.6        | 71.3        | <b>76.9</b> | 70.6        |
| Word±1 | 63.1        | 69.8        | 67.2        | 72.1        | 72.9        | 71.1        |
| Word±2 | 55.3        | 67.7        | 66.8        | <b>74.0</b> | 72.2        | 70.6        |
| Word±3 | 58.0        | 70.5        | 66.8        | 71.7        | 72.3        | 70.6        |
| P&W±1  | <b>72.7</b> | <b>73.9</b> | <b>73.3</b> | <u>73.5</u> | <u>73.4</u> | <b>76.1</b> |
| P&W±2  | 57.7        | 72.3        | 69.2        | 73.5        | 72.2        | 72.1        |
| P&W±3  | 62.0        | 70.4        | 69.6        | 72.1        | 72.6        | 70.6        |

According to the result from C4.5 in Table 2, with data size is not more than 500 sentences, C4.5 makes good accuracy by using only part-of-speech tags with any window sizes. In case of a training data set is equal or more than 1000 sentences, considering only words

give the best accuracy and the suitable window size is depend on the size of training data set. In Table 3, C4.5rule gives high accuracies on considering only part-of-speech tags with any window sizes. In table 4, RIPPER produces high accuracies by investigating only one word and one part-of-speech tag before and after verb-to-be words.

## Conclusion

C4.5, C4.5rule and RIPPER have efficiency in extracting context information from a training corpus. The accuracy of these three machine learning techniques is not quite different, and RIPPER gives the better results than C4.5 and C4.5rule do in a small train set. The appropriate context information depends on machine learning algorithms. The suitable context information giving high accuracy in C4.5, C4.5rule and RIPPER are ±3 words around a target word, part-of-speech tags with any window sizes and ±1 word and part-of-speech tag respectively

This can prove that our approach has a significant in improving a quality of translation. The advantages of our method are 1) adaptive model, 2) it can apply to another languages, and 3). It is not require linguistic knowledge.

In future experiment, we will include other machine learning techniques such as Winnow[1] and increase other context information such as semantic, grammar.

## Acknowledgements

Special thanks to Mr. Sittha Phaholphyayo for marking up the correct meaning of verb-to-be words and Mr. Danooon Nanongkhai, intern student from Computer Engineering Department, Kasertsart University, for his help in testing the experiments.

## References

- [1] Andrew R. Golding and Dan Roth. 1999. A Winnow-Based Approach to Context Sensitive Spelling Correction, *Machine Learning*, Special issue on Machine Learning and Natural Language Processing, Volume 34, pp. 107-130.
- [2] David D. Palmer Marti A. Hearst 1994. Adaptive Sentence Boundary Disambiguation. In the

Proceedings of the Fourth ACL Conference on Applied Natural Language Processing, Stuttgart.

[3] Michael Carl. 1999: Inducing Translation Templates for Example-Based Machine Translation, In the Proceeding of MT-Summit VII, Singapore.

[4] Paisarn Charoenpornasawat., Boonserm Kijirikul. and Surapant Meknavin. 1998. Feature-based Thai Unknown Word Boundary Identification Using Winnow. In *Proceedings of the 1998 IEEE Asia-Pacific Conference on Circuits and Systems (APCCAS'98)*.

[5] Peter F. Brown, John Cocke and etc. statistical approach to machine translation. *Computational linguistics* 16, 1990

[6] Ralf D. Brown 1996. Example-Based Machine Translation in the PanGloss System. In *Proceedings of the Sixteenth International Conference on Computational Linguistics*, Page 169-174, Copenhagen, Denmark.

[7] Ross Quinlan. 1993. C4.5: Programs for Machine Learning Morgan Kauffman.

[8] Virach Sornlertlamvanich and Wantanee Phantachat 1993. Interlingual Expression for Thai Language. *Technical report*. Linguistic and Knowledge Engineering Laboratory, National Electronics and Computer Technology Center, Thailand.

[9] Virach sornlertlamvanich, Tanapong Potipiti and Thatsanee Charoenporn. 2000. Automatic Corpus-Based Thai Word Extraction with the C4.5 Learning Algorithm. *Proceedings of the 18th International Conference on Computational Linguistics (COLING2000)*, Saarbrucken, Germany.

[10] William W. Cohen. 1995 Fast effective rule induction, *In Proceedings of the Twelfth International Conference on Machine Learning*, Lake Tahoe, California, Morgan Kauffman.

[11] Ying Zhang, Ralf D. Brown, and Robert E. Frederking, 2001. Adapting an Example-Based Translation System to Chinese. In *Proceedings of Human Language Technology Conference 2001* p. 7-10. San Diego, California, March 18-21, 2001. *Computational Linguistics*, *Computational Linguistics*, 11/1, pp. 18—27.