A Dependency Parser for Thai

Shisanu Tongchim*, **Randolf Altmeyer**†, **Virach Sornlertlamvanich***, **Hitoshi Isahara**‡

*Thai Computational Linguistics Laboratory
NICT Asia Research Center, 112 Paholyothin Road
Klong 1, Klong Luang, Pathumthani 12120, Thailand
{shisanu,virach}@tcllab.org
†Department of Computational Linguistics and Phonetics
Saarland University, Saarbrücken, 66041, Germany
altmeyer@coli.uni-saarland.de
‡National Institute of Information and Communications Technology
3-5, Hikari-dai, Seika-cho, Soraku-gun, Kyoto, 619-0289, Japan
isahara@nict.go.jp

Abstract

This paper presents some preliminary results of our dependency parser for Thai. It is part of an ongoing project in developing a syntactically annotated Thai corpus. The parser has been trained and tested by using the complete part of the corpus. The parser achieves 83.64% as the root accuracy, 78.54% as the dependency accuracy and 53.90% as the complete sentence accuracy. The trained parser will be used as a preprocessing step in our corpus annotation workflow in order to accelerate the corpus development.

1. Introduction

The research in natural language processing (NLP) for Thai has so far concentrated only within morphological analysis (i.e. word segmentation (Aroonmanakun, 2002; Sornlertlamvanich, 1993; Meknavin et al., 1997; Kawtrakul and Thumkanon, 1997), part-of-speech (POS) tagging (Murata et al., 2002) or both (Kruengkrai et al., 2006)). One of the possible explanations is the lack of syntactically annotated corpora for Thai. Although morphological analysis is a crucial step as the first part of text analysis, the research in higher level, like syntactic analysis, is still important. Some NLP applications, e.g. Machine Translation, require syntactic information and tools for extracting such information from sentences.

When we start a project in developing a machine translation system between Thai and Japanese by using Example-based machine translation (Nakazawa et al., 2006), the development of a syntactically annotated corpus and a parser are necessary. The EBMT system utilizes the dependency structure in aligning the parallel corpus and then extracting the translation examples from the corpus. In this research, we report the results of our statistical dependency parser trained on the preliminary data in our corpus. This parser will help in creating a Thai-Japanese MT prototype and also accelerating the development of syntactically annotated corpus.

2. Dependency Structure in Thai

There are only few studies investigating the dependency parsing for Thai. To our knowledge, the first research regarding dependency analysis was done by Aroonmanakun (1989) in his master thesis. However, this research is based on a very small corpus (50 sentences). The lack of syntactically annotated corpora may be a possible explanation why not much research has been done in this area. Some have been developed, but they are relatively small or not public, for example, a treebank of 400 sentences used in (Satayamas et al., 2005).

The dependency analysis in some languages (e.g. Japanese) considers dependency relations between phrasal units (‘bunsetsu’ for Japanese). In Thai, we consider dependency relations at word level. Figure 1 shows an example of a Thai sentence with dependency relations. The dependency links are drawn from the dependents to their heads. The binary dependency relations between words of Thai can occur in two directions (i.e. left to right and right to left). Unlike some languages, most dependency relations are limited to only one direction (e.g., Japanese (Sekine et al., 2000), Turkish (Eryigit and Oflazer, 2006)). The root node can also be found in arbitrary positions (i.e. at the first, last or the middle of the sentence), while the root node of languages that have a single direction of dependency relations resides at a fixed position. Due to fewer constraints in writing dependency relations and the possible root positions, the search space in finding the correct dependency structure for Thai will be much larger.

In general, constituents in Thai sentences follow the order of Subject-Verb-Object (SVO). However, the word order is more flexible in discourse (Iwasaki and Ingkaphirom, 2005). Since our goal in developing MT project is to handle sentences from the conversation domain, the sentences in the corpus have some features that may be uncommon for other domains, but not for discourse. Iwasaki and Ingkaphirom (2005) outlines three language phenomena that result in the constituent-order variability. The first phenomenon is Zero anaphora. Subjects and objects are often omitted from the sentences, thus the major constituents of some sentences are found with only a verb (V), a verb and an object (VO). The second is the topicalization. Noun phrases functioning as objects are put at the beginning of sentences. Thus, the constituent order like OSV will be possible. The third phenomenon involves the process in adding a constituent later than its usual position. Thus, some sentences

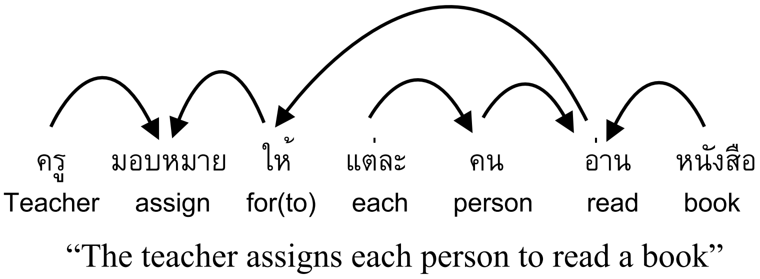

Figure 1: An example of a Thai sentence with dependency relations.

with the constituent order like VS or VOS can be found. These ambiguities are challenging for the development of a Thai parser for this domain.

3. Parser

Machine learning algorithms have been applied to natural language parsing. Some studies adopted shift-reduce parsers which require proper sequences of actions as instructions for the parsers in processing the sentences. Machine learning algorithms have been used to determine the sequences of actions in parsing given sentences (e.g., support vector machines (SVMs) (Yamada and Matsumoto, 2003; Cheng et al., 2005), Maximum entropy models (Ratnaparkhi, 1999)). Another approaches use machine learning algorithms to estimate the probability that one word modifies another word. The probability model can be represented as the probability matrix showing the probability values of binary dependency relations. Some machine learning algorithms used in estimating the probability values are SVMs (Kudo and Matsumoto, 2000) and maximum entropy models (Uchimoto et al., 1999; Sekine et al., 2000). In this work, we adopt SVMs to estimate the probability values. Since the root node for Thai sentences can be found in arbitrary positions, the root node identification will be considered as a separated process. Some studies (Isozaki et al., 2004) also separated the root node identification from the main parsing task. The parsing process of our parser has three main steps:

1. Root node identification: We train an SVM to estimate the probability value of each word for being the root node. The word with the highest probability value is selected as the root node. Seven features are used in the root model:
   - POS
   - Relative position
   - Number of verbs
   - Number of equivalent POS in front of this word
   - Number of equivalent POS after this word
   - Number of equivalent major POS in front of this word
   - Number of equivalent major POS after this word

   Note that some sentences in the corpus may not have any verb (e.g. a short answer for a question). These sentences may consist of a noun with some function words. Thus, the root node in this case is a noun, rather than a verb as usual.

2. Build the dependency matrix: The second model based on an SVM is used to derive the probability of dependency relations. The features are as follows:
   - POS of the first word
   - POS of the second word
   - Direction of dependency relations (left to right, right to left)
   - Distance between both words
   - Major category of the first word (function word, content word)
   - Major category of the second word (function word, content word)
   - Major POS of the first word
   - Major POS of the second word
   - Relative position of the first word
   - Relative position of the second word

3. Find the best dependency structure: Given the prospective root position and the dependency matrix, the final step is to find the best dependency structure. The search algorithm is based on a beam search with maintaining the top k candidates during the search process. We define a sentence S as a sequence of words $\{w_1, w_2, ..., w_n\}$. Let $Dep(i) = j$ mean the word w_i modifies the word w_j (w_j is the head of w_i). The probability $Prob_{root}(i)$ means the probability of w_i being the root node of the sentence, while $Prob_{dep}(i, j)$ is the probability that w_i modifies w_j. We define the head of the root node as -1 and $Prob_{dep}(Root, -1) = 1$.

   The problem is to find a dependency structure D ($\{Dep(1), Dep(2), ..., Dep(n)\}$) that maximizes the conditional probability. Assume that the probability values of dependency relations are mutually independent. We can calculate the probability of each dependency structure as follows:

   $$P(D|S) = \prod_{i=1}^{n} Prob_{dep}(i, Dep(i))) \qquad (1)$$

   For the sake of simplicity, we illustrate the search algorithm by using $k = 1$. The search algorithm is outlined in Algorithm 1. The beam search follows the

Algorithm 1: FINDDEPENDENCYPATTERN

```
$\mathcal{T}_s \leftarrow \{w_1, w_2, ..., w_n\}$
$\mathcal{T}_u \leftarrow \{\}$
$l = \text{argmax}_{i \in [1,n]} Prob_{root}(i)$
Remove $w_l$ from $\mathcal{T}_s$
Add $w_l$ to $\mathcal{T}_u$
while $\mathcal{T}_s$ is not empty do
    Find $w_o$, $w_o \in \mathcal{T}_s$ and its head $w_p, w_p \in \mathcal{T}_u$ which maximize the conditional probability
    Remove $w_o$ from $\mathcal{T}_s$
    Add $w_o$ to $\mathcal{T}_u$
end
```

same procedure but maintaining the top k candidates during the search process.

Normally, SVMs provide the classification of the input instance. To obtain the probability values from SVMs, we use the probability estimation in LIBSVM (Chang and Lin, 2001).

4. Experimental Results

The experiment is done on the first portion of annotated corpus. The corpus consists of 2692 sentences. The sentence length ranges between 2 words to 20 words with an average of 5.70. Although the sentences in the corpus seem to be short, we argue that sentences in the conversation domain impose several ambiguities and the search space for Thai is larger than some languages due to the lack of constrains.

The parser finds the dependency patterns that follow the projectivity assumption. That is, the dependency relation does not cross another dependency relation and no dependency relation covers the root node. The gold standard of POS tags are used.

For the SVMs, we use the RBF kernel with $(C = 1, \gamma = 0.14)$ for the root identification and $(C = 1, \gamma = 0.1)$ for the dependency analysis. The beam width parameter of the beam search is set to 3. Three performance metrics are used:

- *Root accuracy*: This metric represents a portion of sentences with correctly identified roots.
- *Dependency accuracy*: The dependency accuracy determines a ratio of correct dependency relations to all dependency links.
- *Complete sentence accuracy*: The last metric shows a portion of sentences with correct roots and dependency patterns.

The corpus is divided into 2423 sentences as the training set and 269 sentences as the test set. The performance of our parser is as follows:

- Root accuracy = 83.64%
- Dependency accuracy = 78.54%
- Complete sentence accuracy = 53.90%

Note that we cannot compare the performance with a baseline like some studies (Eryigit and Oflazer, 2006; Uchimoto et al., 1999) since the dependency relations of Thai are two directions and the root positions are not fixed. To judge how well our parser performs, we survey some similar works in developing dependency parsers from the previously published literature (see Table 1). We acknowledge that the results of these studies are not directly comparable with each other or our results since the experiments have been done on different data and languages. To a certain extent, however, these results provide some insights of how other proposed parsers perform. In terms of the complete sentence accuracy, our results are acceptable comparing with the results of several parsers. Our dependency accuracy is relatively low. One of the reasons may come from the difficulties in finding the right root nodes of our parser. The incorrect root nodes will affect the performance of the later steps in finding the dependency structure.

5. Conclusions

This work presents some preliminary results in developing a dependency parser for Thai. The parser is composed of three components. The first one is the root identification. The second one is the dependency analysis. The last one is the search algorithm based on the beam search. The first two components utilize SVMs to estimate the probability values. Although the parser has been tested and trained on a very small corpus, the development of this parser is still important as a tool for accelerating the corpus development.

Acknowledgment

This work was done while the second author spent two months at Thai Computational Linguistics Laboratory.

6. References

Wirote Aroonmanakun. 1989. A dependency analysis of thai sentences for a computerized parsing system. Master thesis, Department of Linguistics, Chulalongkorn University.

Wirote Aroonmanakun. 2002. Collocation and Thai word segmentation. In *Proceedings of the 5th SNLP & 5th Oriental COCOSDA Workshop*, pages 68–75.

Chih-Chung Chang and Chih-Jen Lin, 2001. *LIBSVM: a library for support vector machines*. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

Yuchang Cheng, Masayuki Asahara, and Yuji Matsumoto. 2005. Chinese deterministic dependency analyzer: Examining effects of global features and root node finder. In *Proceedings of the Fourth SIGHAN Workshop on Chinese Language Processing*, pages 17–24.

Gülsen Eryigit and Kemal Oflazer. 2006. Statistical dependency parsing for turkish. In *EACL*. The Association for Computer Linguistics.

Hideki Isozaki, Hideto Kazawa, and Tsutomu Hirao. 2004. A deterministic word dependency analyzer enhanced with preference learning. In *COLING '04: Proceedings of the 20th international conference on Computational Linguistics*, page 275, Morristown, NJ, USA. Association for Computational Linguistics.

Table 1: Parser accuracy for various languages in the previously published studies

Language	Root accuracy	Dependency accuracy	Complete sentence accuracy
English (source: (Isozaki et al., 2004) (Nivre and Scholz, 2004)(Yamada and Matsumoto, 2003))	84.3-95.7%	87.3-91.2%	30.4-40.7%
Japanese (source: (Kudo and Matsumoto, 2002) (Sekine et al., 2000)(Uchimoto et al., 1999))	NA	87.14-90.46%	40.60-53.16%
Turkish (source: (Eryigit and Oflazer, 2006))	NA	73.5%	38.7%
Chinese (source: (Cheng et al., 2005))	90.94%	86.18%	61.33%

Shoichi Iwasaki and Preeya Ingkaphirom, 2005. *A reference grammar of Thai*, chapter 30, pages 374–376. Cambridge University Press.

Asanee Kawtrakul and Chalatip Thumkanon. 1997. A statistical approach to Thai morphological analyzer. In *Proceedings of the 5th Workshop on Very Large Corpora*, pages 289–296.

Canasai Kruengkrai, Virach Sornlertlamvanich, and Hitoshi Isahara. 2006. A conditional random field framework for Thai morphological analysis. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation*.

Taku Kudo and Yuji Matsumoto. 2000. Japanese dependency structure analysis based on support vector machines. In *Proceedings of the 2000 Joint SIGDAT conference on Empirical methods in natural language processing and very large corpora*, pages 18–25, Morristown, NJ, USA. Association for Computational Linguistics.

Taku Kudo and Yuji Matsumoto. 2002. Japanese dependency analysis using cascaded chunking. In *COLING-02: proceeding of the 6th conference on Natural language learning*, pages 1–7, Morristown, NJ, USA. Association for Computational Linguistics.

Surapant Meknavin, Paisarn Charoenpornsawat, and Boonserm Kijsirikul. 1997. Feature-based Thai word segmentation. In *Proceedings of NLPRS97*, page 289296.

Masaki Murata, Qing Ma, and Hitoshi Isahara. 2002. Comparison of three machine-learning methods for thai part-of-speech tagging. *ACM Trans. Asian Lang. Inf. Process.*, 1(2):145–158.

Toshiaki Nakazawa, Kun Yu, Daisuke Kawahara, and Sadao Kurohashi. 2006. Example-based Machine Translation based on Deeper NLP. In *Proc. of the International Workshop on Spoken Language Translation*, pages 64–70, Kyoto, Japan.

Joakim Nivre and Mario Scholz. 2004. Deterministic dependency parsing of english text. In *COLING '04: Proceedings of the 20th international conference on Computational Linguistics*, page 64, Morristown, NJ, USA. Association for Computational Linguistics.

Adwait Ratnaparkhi. 1999. Learning to parse natural language with maximum entropy models. *Machine Learning*, 34(1-3):151–175.

Vee Satayamas, Chalatip Thumkanon, and Asanee Kawtrakul. 2005. Bootstrap cleaning and quality control for thai tree bank construction. In *The 9th National Computer Science and Engineering Conference*, Bangkok, Thailand, Oct 27–Oct 28. (In Thai).

Satoshi Sekine, Kiyotaka Uchimoto, and Hitoshi Isahara. 2000. Backward beam search algorithm for dependency analysis of japanese. In *COLING*, pages 754–760. Morgan Kaufmann.

Virach Sornlertlamvanich. 1993. Word segmentation for thai in machine translation system. Machine Translation, National Electronics and Computer Technology Center, Bangkok.

Kiyotaka Uchimoto, Satoshi Sekine, and Hitoshi Isahara. 1999. Japanese dependency structure analysis based on maximum entropy models. In *EACL*, pages 196–203.

H. Yamada and Y. Matsumoto. 2003. Statistical dependency analysis with support vector machines. In *The 8th International Workshop of Parsing Technologies (IWPT2003)*, pages 195–206.