Development of Vietnamese Parser based on Phrase Pattern Grammar

Hong Luong Thi Bich  Nguyen Thanh Hung  Hideto Ikeda
Ritsumeikan University
{hong, hungnt, hikeda}@is.ritsumei.ac.jp

Abstract
Phrase Pattern Grammar (PPG) [14] is a phrase structure grammar specified by 3-tuple (Phrase Patterns, Vocabulary, Embedding Rules among Phrases and Vocabulary). In PPG, a sentence can be expressed by a tree in which each node is a phrase and each path is corresponded by a rule. In this paper, using PPG we proposes an efficient Vietnamese parser which implements a top-down algorithm, called peeling algorithm, depth-first tree search, restriction of child nodes by week-constraint for embedding rules and search priority of child nodes by length of embedded pattern.

1 Introduction
Syntactic analysis, called parsing, is the process of structuring a linear representation in accordance with a given grammar [1] and considered as a central problem and a challenge in the field of natural language processing [19]. The conventional parsing methods and techniques are focusing on determining grammatical structure of sentences with respect to a given formal grammar [3].

There exist several parsers which use phrase structure grammar [4, 5, 6]. The phrases have a hierarchical structure and contain sub-phrases [20]. These parsers could bring more robustness and accuracy than the ones using dependencies such as Mini-par [7], or the Link Parser [8]. However, there are not many effective parsers that have been successfully applied to Vietnamese, since the word order in Vietnamese is very complicated more than English. There are currently some Vietnamese parsers that use HPSG [9] or PCFG model [10]. They are strongly depending on word segmentation. The efficiency is low due to limitation of Vietnamese Tree bank.

Ikeda et al. 2010[14] proposed a new language model based on Para-Phrase Grammar (PPG) that is a grammar consisted of the set of multilingual phrases and embedding rules of phrases into phrases. These para-phrases cover a word, a grammatical phrase, a clause and the parent sentence itself, that is a component of sentence.

This paper proposes a top-down algorithm, called peeling algorithm on the base of PPG. It implements a Vietnamese parser which allows each sentence pattern in Vietnamese to be translated as a sequence of phrase patterns which easily correspond to the same meaning patterns in other languages. The paper is organized as follow; Section 2 defines the proposed PPG. Section 3 discusses problems when applying PPG to Vietnamese sentences and proposes a peeling algorithm. Experiment result is shown in section 4. Section 5 is conclusion and future works.

2 Phrase pattern Grammar

2.1 Definition of phrase and phrase pattern

Even if we use morpheme-level or word-level knowledge managed independently, like WordNet, it is difficult to analyze sentence structure syntactically and semantically because the decomposition of sentences into lists of morphemes or words may destroy the semantics of phrases and sentence [15, 16, 17]. It is therefore recommended to use phrase-level semantics or knowledge to analyze the semantics of a sentence [2].

In PPG, we normalize all POS (Part Of Speech) into three types of phrases: noun phrase (N), verb phrase (P) and sentence phrase (S). We shall define the para-phrase pattern (or phrase pattern) as follow;

Definition 1: A para-phrase pattern (or phrase pattern) in this paper is a string made by a phrase of which is noun, predicate and sentence phrases are replaced by the variable “_”.

For example, the sentence:

“Unlike English, Vietnamese is a monosyllabic language.”

(2.1a)

has the following phrases conventionally: “Unlike English[PP]; Vietnamese[NN]; a monosyllabic language[NP]; is a monosyllabic language[VP]; Vietnamese is a monosyllabic language[S]; Unlike English, Vietnamese is a monosyllabic language[S]”. 
In PPG, we only have following phrases: “S0=Unlike-,-,-is-a-_.([N1],[N2],[N3]); N1= English, N2=Vietnamese, N3= monosyllabic language”. The “-“ (hyphen) symbol is stand for space between syllables.

The above approach can be applied to other languages such as Vietnamese without differences. As an example, figure 1 depicts an aligned sentence between English and Vietnamese. It shows that there is a natural correspondence between functions of the two languages. In fact, it is possible to make a correspondence among not only English and Vietnamese, but also other languages [21, 22, 23]. We have already investigated this issue for Japanese, Korean, and Chinese.

2.2 Constraints of phrase embedding

In order to reconstruct a correct sentence, it is necessary to establish the set of embedding rules of phrase patterns. Although we can avoid some of neither illegal phrases nor sentences by using these types, it is not enough to avoid for making incorrect sentences. It is necessary to establish the set of rules to control the correctness of sentences. In conventional grammars, the rules are described by using part-of-speech (POS). But by a POS-based approach, we cannot avoid exceptions. Another approach is to assign a concept code to each parameter of each function. Typical examples of concept code are person-name, organization-name, food, place and so on. This concept code approach for embedding control is better than POS control. It is, however, not perfect also. We adopt concrete-phrase (CP) approach in which each combination of phrases is controlled to be possible to embed or not.

For example an English function:

S0=Unlike-,-,-is-a-_.([N1],[a-object:N2],[main: N3]); [ appearance]

where, a-object code is specify that which has a particular attribute, “appearance” code is about condition and comparison.

3 Vietnamese Parsing based on Phrase Pattern Grammar

3.1 Current problems of Vietnamese parsing

Research on Vietnamese parsing encounters the difficulties with segmentation, ambiguity, data insufficiency and translation incapability. Since Vietnamese is monosyllabic language, accurate segmentation is challenging. One word can be combined by more than one syllables separated by space character, e.g., “chúột” (mouse), “bàn phím” (keyboard). “máy vi tính” (computer). Vietnamese has many ambiguous sentences, which burden the design of syntax parsing [9]. Moreover, with the missing of linguistic data corpora, high accuracy parsing model could not be built. The translation therefore depends much on accuracy of the lexical analysis, parsing and correct syntax structure.

3.2 Database organization

The database for Vietnamese Parsing based on PPG contains 3 tables: V-word, V-func, V-phrase

• V-word table stores all the words in dictionary used to build Vietnamese sentences. This table also stores POS, search keyword, word type, meaning, ...
• V-func table uses S, N, P to present Vietnamese functions following PPG’s rule.
• V-phrase table stores phrases which require accurate translation or phrases usually used with each other. The phrases are stored with their function structure and directly use in parsing.

3.3 Parsing algorithm

Our parser is conceptualized to consist of 4 steps. We shall demonstrate the algorithm in the figure 2,3,4,5 and as follows:

• Step-1 (figure 2): Segmentation & POS tagging.

The parser looks up the database (including V-word, V-phrase) to choose the correct combination of symbols for the sentence to be parsed. At the end of the first step, the parser would not only segment the input sentence into a list of words and phrases with POS tags, but would also associate each word/phrase with a list of potential functions. For example as shown in figure 2, the step-1 segments the Vietnamese source sentence to a sequence of Vietnamese words and phrases, then look up the database for potential functions. The result after this step is store in “Function list”. The real number of functions appear in this list may reach to hundreds according to the size of the databases.

• Step-2 (figure 3): Prediction of skin functions.

The parser analyzes segmentation & POS tagging results and predicts skin functions for faster and better parsing. Because the parser is implemented by a top-down algorithm (peeling algorithm), it is necessary to reduce the redundant backtrack by finding the right skin function of the sentence since early stages of parsing.

• Step-3 (figure 4): Priority analysis for function list. The parser analyzes the probability of each function in potential functions list by considering their parameters.
S=w“Nếu anh không thoái mái với chờ ở đó, sao anh
không chuyển đi chờ khác?” (If you are not comforta-
ble with the house, why don’t you change it?)

Figure 1: Function list

Figure 2: Segmentation & POS tagging

Figure 3: Prediction of skin functions

Figure 4: Priority analysis for function list

Figure 5: Recursion

<table>
<thead>
<tr>
<th>Skin list</th>
<th>Function list</th>
<th>Priority list</th>
</tr>
</thead>
<tbody>
<tr>
<td>S= Nêu...?([S][S]);</td>
<td>vWord dic</td>
<td>vPhrase function dic.</td>
</tr>
<tr>
<td>vWord dic</td>
<td>vPhrase function dic.</td>
<td>Function list</td>
</tr>
<tr>
<td>Segmentation</td>
<td>(1)</td>
<td>Function list</td>
</tr>
<tr>
<td>NêuC/P</td>
<td>NULL</td>
<td>P=không-_(P);</td>
</tr>
<tr>
<td>không/R</td>
<td>S=không-_(N],[P]);</td>
<td>P=thoai-mái(P);</td>
</tr>
<tr>
<td>thoai-mái/A</td>
<td>P=thoai-mái-_(N);</td>
<td>P=thoai-mái-_(N);</td>
</tr>
<tr>
<td>vôi/C</td>
<td>P=với-_(N);</td>
<td>P=với-_(N);</td>
</tr>
<tr>
<td>chở-ô-dỏ/N</td>
<td>P=chở-ô-dỏ(P);</td>
<td>P=chở-ô-dỏ(P);</td>
</tr>
<tr>
<td>chuyển-di/V</td>
<td>P=chuyển-di-_(N);</td>
<td>P=chuyển-di-_(N);</td>
</tr>
<tr>
<td>chữa-khắc/N</td>
<td>P=chữa-khắc(P);</td>
<td>P=chữa-khắc(P);</td>
</tr>
<tr>
<td>P=không-_(P);</td>
<td>P=không-_(P);</td>
<td>P=không-_(P);</td>
</tr>
<tr>
<td>POSITION</td>
<td></td>
<td></td>
</tr>
<tr>
<td>vPhrase Dictionary</td>
<td>22,620 phrase functions</td>
<td></td>
</tr>
<tr>
<td>vPhrase Dictionary</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Table 1: Experiment environment</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sc 1450 sentences</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sa 1500 sentences</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sp 1400 sentences</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Precision Recall F-score</td>
<td></td>
<td></td>
</tr>
<tr>
<td>93.3% 96.6% 94.9%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Conclusion</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The goal of this step is to reconstruct the parsing tree
that allows the parser to retrieve the highest probability
function firstly.

- Step-4 (figure 5): Recursion (Depth-first-
search). The parser uses the depth-first parsing tree
search algorithm to recursively select from skin func-
tions towards leaf functions until the sentence is
completely parsed. If suitable leaf functions are not
found, the parser must backtrack - that is, return to
one higher tree node and select another function.

4 Experiments

This section presents experiments and results of pars-
ing for Vietnamese sentences on the part-of-speech
tagged section of the Vietnamese Treebank. The
Treebank is currently composed of 10409 sentences
which are manually segmented, POS tagged and
parsed. We take a pre-process the Treebank into
three set of data: word dictionary, function dictionary
and phrase dictionary. Using the pre-processed data
above, the proposed PPG model is implemented using
PHP programming language.

To evaluate the performance, we use recall (R),
precision (P) and F parameters, which can be defined
as follows:

\[
R = \frac{Sp}{S_p} \quad P = \frac{Sp}{S_a}
\]

F = \frac{2}{\frac{1}{R} + \frac{1}{P}}

The experiment environment is shown in table 1. Thus, we have the result indicated in table 2.

Table 1: Experiment environment

Table 2: Experiment result

5 Conclusion

In this paper, using PPG we adopt an efficient
Vietnamese parser which is implemented by a top-
down algorithm, called peeling algorithm, depth-first
tree search, restriction of child nodes by week-
constraint for embedding rules and search priority of
child nodes by length of embedded pattern. The ap-
proach can resolve word ambiguity and segment the
input sentence into phrase-functions. Experimental
results show a result with an F-score of 94.9%, high-

Copyright(C) 2012 The Association for Natural Language Processing.
All Rights Reserved
er than the results of existing publicly available Vietnamese word segmentation systems.

Since the number of current Vietnamese sentences used for the Treebank in the experiment is small, the number of para-phase functions is also limited. The phrase function dictionary needs to be extended more. It will lead to the management job for the huge phrase function database in the future. Furthermore, to convert to other languages, we need an algorithm to align between phrase patterns of different languages.

References


[19] Anh-Cuong Le, Phuong-Thai Nguyen, Hoai-Thu Vuong, Minh-Thu Pham, Tu-Bao Ho. An Experimental Study on Lexicalized Statistical Parsing for Vietnamese. 2009 International Conference on Knowledge and Systems Engineering


