Exploiting the Syntax-Annotated Corpus for Analysing Vietnamese Syntax

Phan Thi Ha, Ha Hai Nam, Đo Xuan Cho

Abstract: ***This paper presents an algorithm for automatic extraction of the PCFG (probability context free grammar) from Viettreebank and an algorithm for constructing the Vietnamese parser based on the PCFG for Vietnamese sentence analysis. The parsing algorithm for each sentence is developed from the jurafsky and martin algorithm [5]. Applied to the Vietnamese language, an input sentence is labeled by an available part-of-speech(POS) tagging tool, while for Jurafsky and Martin , the input sentences is unlabeled POS of which words are separated by white space.***

Keywords: CFG, PCFG, CYK, PCYK, Treebank, Probability Context Free Grammar, parser

I. INTRODUCTION

Parsing is an important step in natural language processing, with a high quality parser that improves the efficiency of natural language processing systems such as machine translation, document summarization, and Q & A systems.

For Vietnamese, all parsers need the Vietnamese syntax rule, or grammar for Vietnamese, which is represented by a specific grammatical system. This set of rules can be obtained from some of the materials developed in the KC01.01 / 06-10 project, the VietTreebank, a syntax annotation. In VietTreebank, a group of linguistic specialists performed a syntactic information annotation for a Vietnamese text archive and the component annotation format was encoded in brackets. The corpus is divided into three volumes corresponding to three levels of labeling, word segmentation, part-of-speech tagging and syntax tagging. The syntax labeling package includes 10471 sentences (225085 vocabulary words). The length of sentences ranged from 2 to 105 words, with an average length of 21.75 words. There are 9314 sentences (occupying 88.95%) of the maximum length of a sentence has no more than 40 words. The syntax trees have a height in the range of 5 to 10, most commonly 7 (1436). Detailed information about VietTreebank is provided in the document [1].

Based on the syntax information in VietTreebank the article introduces the construction of the grammar rule for the Vietnamese parsing problem. This paper presents the development of the Probability Context Free Grammar (PCFG) algorithm from VietTreebank with experiments and evaluation. At the same time, the Vietnamese parser on the PCFG grammar was developed, in which the parsing algorithm for each sentence was improved from Jurafsky and Martin's PCYK algorithm [5].

Revised Version Manuscript Received on October 06, 2017.

Phan Thi Ha, Posts and Telecommunications Institute of Technology, Hanoi, Vietnam, South China. E-mail: hapt@ptit.edu.vn, hathiphan@yahoo.com

Ha Hai Nam, Posts and Telecommunications Institute of Technology, Hanoi, Vietnam, South China. E-mail: namhh@ptit.edu.vn

Đo Xuan Cho, Posts and Telecommunications Institute of Technology, Hanoi, Vietnam, South China. E-mail: chodx@ptit.edu.vn

With the approach for Vietnamese language, an input sentence is part-of-speech tagging by the available tool, while for Jurafsky and Martin , the input sentences is ***unlabeled*** part-of-speech of which words are separated by white space . After POS tagging, the probability category P(POS|word) in a grammar that does not affect finding the most probable analysis tree (as evidenced in section 3), in the parsing algorithm the author's PCYK was adjusted by substituting the probability P(POS|word)=1 instead of calculating probability in VietTreebank.

The layout of the paper is divided into the following sections: Section 1 is introduction, Section 2 discusses the PCFG grammar extraction algorithm from VietTreebank, Section 3 introduces Vietnamese parser with PCFG grammar, Section 4 shows test and evaluation, Section 5 is the conclusion.

II. ALGORITHM FOR EXTRACTING SENTENCE STRUCTUER GRAMMAR OF VIETTREEBANK

The automatic extraction of CFG grammar rules, PCFG from syntax tree bank for parsing have been used for many languages, such as English [5], German, ... For Vietnamese language, the issue of exploiting VietTreeBank for research and development of language processing applications for Vietnamese is developed [1, 5]. The paper presents the method of extracting PCFG rules from the Vietnamese language bank named VietTreebank. The method used here is similar to that of Roberto Valenti, Jurafsky and Martin [5]. However, the detailed algorithm in each step of the method was built from scratch. Basically, the extraction process is done through the following steps:

Step 1. Extract the rules of the CFG grammar

This step will allow the extraction of Context Free Grammar rules (CFGs) from VietTreebank in parentheses and also eliminate the rules for generating the term (for example, Np-> Lan). VietTreebank will read each file and split up into sentences, rules of each sentence is extracted by using **Algorithm 1.**

Algorithm 1 uses the Stack to cache the extraction rules from each tree of analysis, the complexity of the algorithm is O (n + m), where n is the number of partial labels (sentences, phrases, POS) corresponding to the number of vertices in the graph, m is the relationship between the partial labels correspond to the number of edges in the graph.

In the algorithm, each element in the Rulebeans [] array is a RuleBean consisting of two components:

String left: label to the left of the rule

String right: the labels to the right of the rule, separated by commas.

NewRule and *rulTop*: These are two rules used to mark newly created rules and rules on the top of the stack.

For example, with a syntactical comment: "Nguyen Thanh My never told me that he loves water"

<S>

(S (NP-SUB (Nguyen Thanh My))

(VP (R)) (P never) (V say)

(PP-DOB (E with) (P i))

(C is)

(SBAR-DOB (NP-SUB))

(VP (love) (N country)))))

(,,))

After extraction, the obtained rules (after eliminating the rules that produce the summation sign, for example, the rule R-> yet, P-> ever) would be:

NP-> Np; PP-> E | P;

NP-> N; VP-> V | N;

SBAR-> NP | VP;

VP-> R | P | V | PP | C | SBAR;

S-> NP + VP

Step 2. Transform CFG into probabilistic non-context (PCFG)

The probability of each rule has format A→ α (with A being non-terminating, α is any non-empty character) is computed by (2.1)

$$P(A\rightarrow\alpha) = \frac{frequency(A->\alpha)}{frequency(A)} \quad (2.1)$$

Step 3. Turn PCFG into PCFG in standard Chomsky format

The PCYK parsing algorithm [4] is only applicable to Chomsky (CNF) standard with PCFG rules. So in this step it is necessary to convert the PCFG rules into the rules of the PCFG grammar in the CNF form, where each rule can only be in one of the forms A-> *a* or A-> BC, where *a* is the ending sign, A, B, C are non ending symbols.

The rules are obtained from the grammar VietTreebank in step 2 will no longer form A-> a, or A-> ∅ anymore but only form A-> B (probability *p*) or A-> BCDE ... (*p*)

- If the rule of the form A-> BC (*p*) is the same

- If the rule A-> B (*p*) is always converted to A-> B∅ (*p*).

- If the rule is form A-> BCDE .... (*p*) then it is converted to A-> BC '

Algorithm 1. Rulebeans[]RulbuildRules**(String S)**
input : a tree of VietTreebank
ouput : List of obtained rules (RulBeans)

```
1.  {
2.  Rulbeans[]=ϕ;
3.  stack=null;
4.  for (i= first bracket;i<= end bracket; i=next bracket)
5.  {
6.  if (i =="("){
7.  left←first_token from i to next bracket;
8.  delete(function lable) in left;
9.  if (stack==null){
10.   newRule.left←left ;
11.   push(stack,newRule);}
12. else {
13.   ruleTop ← pop(stack)
14. if (ruleTop.right=∅)
15.     ruleTop.right←left
16. else
17.     ruleTop.right← ruleTop.right+"|" left;
18.   push(stack,ruleTop);
19.   newRule.left ← left;
20. push(stack, newRule);
21.  }
22. else {
23.    if((the sequence between the front and ith parenthesis )!=Null)
24.             pop(stack);
25.       else
26.             Rulbeans[]← pop(stack);
27.
28. }
29. }
30. return(Rulbeans)}
```

III. PARSE VIETNAMESE USING PCFG GRAMMAR

One approach to building a parser is to use statistical methods. This method will use the rules in the Context Free Grammar (CFG) grammar along with the probability information of each law (called Context Free Grammar of probability - PCFG) to provide a parsed tree has highest probability for each input sentence. This choice minimizes ambiguity over Context Free Grammar when parsing any sentence.

The parser accepts input as any sentence; the user output can choose the default tree of the greatest probability analysis or all the trees of analysis which have the same probability information attached. By using Jurafsky and Martin method, the process is divided into two phases:

Stage 1: Word Separation (words are separated by white space).

Stage 2: Parsing the sentence which words are separated by white space, the parser will parse the sentence into possible syntactic trees and the highest probability tree will be given preference.

For Vietnamese, this word separation is very complicated because there are double words (polyphonic), which is different from English or other languages. The word separation is simply based on the space. At present, KC01 / 01 has provided some pre-treatment tools for Vietnamese, such as word separation, Part of Speech (PoS) tagging, etc. Based on the characteristics of Vietnamese and Martin's method, we use Part of Speech tagging tools for Vietnamese sentence (at the same time word separation) for parsing. This approach will also minimize the ambiguity in the analysis tree generation compared to Jurafsky and Martin's approach. Processing is done in two phases:

Phase 1: Instead of word separation, always assign POS (simultaneous word separation) by using the label toolkit from the existing type. The POS tagging tool of the Vietnamese type authored by Le Hong Phuong has been used here [7].

Phase 2: The parsing of the sentence has been separated word and labeled POS, the parser will parse the sentence into possible syntactic trees and the highest probability tree will be given preference.

According to this approach, the selection of the highest probability tree depends only on the probability of the rules ($L_h \rightarrow R_h$) (with L_h, R_h is non-terminating nodes) independent of the probability P (PoS of word | word).

Prove:

The probability of combining a particular T-parse tree and one S-sentence has been separated word and labeled POS from the type defined as P (T, S),

P (T, S) = P (T) (3.1)

Since P (T, S) = P (T) * P (S|T) and P (S | T)

According to [5] $P(T_k, S) = \prod_{i=1}^{n} P(L_i \rightarrow R_i)$ (3.2), where n is the number of rules that make up a T tree, T_k is an arbitrary tree of statements S, L_i and R_i are the left and right sides of the rule i in grammar.

Suppose sentence S has n_1 word => after word segmentation and part of speech taging have n_1 rules ($t=1 \div n_1$). Set $n_2 = n-n_1$

$$\Rightarrow P(T_k,S) \overset{(3.2)}{=} \prod_{t=1}^{n_1} P(TL_t \rightarrow word_t) * \prod_{h=1}^{n_2} P(L_h \rightarrow R_h) \quad (3.3)$$

The combined probability of analysis trees for a particular sentence will be calculated and selected for the largest probability tree ($P(T_k |S)$). Call T' is an analysis tree satisfying $\arg\max_{T_k} P(T_k|S)$, on the other hand we have

$\Rightarrow P(T_k|S) = \frac{P(T_k,S)}{P(S)}$ for each tree S, P(S) is a constant (3.4).

$$\Rightarrow \arg\max_{T_k} \frac{P(T_k,S)}{P(S)} \text{ Depends only on } \arg\max_{T_k} P(T_k,S) \quad (3.5)$$

$P(T_k,S) = \prod_{t=1}^{n_1} P(POS_t \rightarrow word_t) * \prod_{h=1}^{n_2} P(L_h \rightarrow R_h)$, where $P(POS_t \rightarrow word_t)$ of each rule ($POS_t \rightarrow word_t$) is constant in grammar (calculated from Treebank).

Table 4. 1. Number of Obtained Rules

Testing data	Size of testing data (KB)	Ratio of testing data and VietTreebank	Number of CNF rules	Recall	Precision
F1	129	0.019	1274	0.687	0.642
F2	137	0.020	1325	0.659	0.683
F3	169	0.024	1884	0.755	0.767
F4	222	0.032	1866	0.719	0.713
F5	413	0.060	3007	0.769	0.76
F6	800	0.115	5271	0.781	0.799
F7	867	0.125	6409	0.754	0.660
F8	2683	0.386	15696	0.784	0.690
F9	5478	0.789	25560	0.798	0.696
F10	6011	0.866	27577	0.818	0.715

=> $\arg\max_{T_k} \frac{P(T_k,S)}{P(S)}$ Depends only on

In other words, the selection of the highest probability tree only depends on the probability of the rules ($L_h \rightarrow R_h$) (with L_h, R_h is non-terminating nodes)

So for simplicity and not reducing the generality for finding the highest probability tree using Martin's PCYK algorithm, we replaced algorithm 2 (replacing $P(POS_t \rightarrow word_t)=1$ in line 3 of the algorithm), because this eliminates the need to save the ($POS_t \rightarrow word_t$) rules and also does not have to calculate $P(POS_t \rightarrow word_t)$ => no need to extract rules of form ($POS_t \rightarrow word_t$), these rules will be removed during extracting the CFG rules from Treebank, this is shown in algorithm 1. This approach also minimizes ambiguity in the parsing tree's output.

Algorithm 2 Function PCYK ()

Input:

- The string of words have been separated word and labeled POS from the type saved in the words [] array.
- The rules (grammar) and the corresponding probability of PCFG grammar

Output:

- Analytical tree has the highest probability and probability of the tree

1. **for** (j=1 →LENGTH (*words*))
2. **for** all{A|A->*words*[j]∈*grammar* }
3. *table*[*j-1,j,A*] ← 1
4. **for** (*i= j-2→0)*
5. **for** (*k = i+1→j-1*)
6. **for all** {*A*|*A*→*BC* ∈*grammar***and***table*[*i,k,B*]>0 and *table*[*k,j,C*]>0}
7. **if**(*table*[*i,j,A*]<*P*(*A*→*BC*)x*table*[*i,k,B*]x*table*[*k,j,C*]) *table*[*i,j,A*]←*P*(*A*→*BC*)x*table*[*i, k, B*]x *table*[*i,j,C*] *back*[*i, j, A*] ← {*k, B, C* };
8. **return** BUILD-TREE(***back*[1,** LENGTH(***words***),S]), *table*[1, LENGTH(***words***), S];

IV. TESTING AND EVALUATION

Parsing software based on the PCFG grammar has been developed by the authors, including two main functions, function 1, automatic extraction of PCFG grammar rules VietTreebank, function 2, sentence structure analyze method for any input sentence based on the rule of the PCFG grammar.

The extracted results are shown in Table 4.1, training data is divided into regions, where F1, F2, F3, F4, F7, F8 are non-overlapping data regions and subsets of F9⊂F10 with different sizes is taken from VietTreebank (6.78MB).

The results show that the number of CNF rule increases with the size of Viet. Treebank proving VietTreebank is not big enough. The accuracy of the analyzer is calculated as follows: The accuracy is based on the parentheses (i, j, component labels) of the output tree in brackets, where:

Recall = (Number of correct parentheses of the generated tree) / (number of correct parentheses of the standard tree);

Precision = (Number of correct parentheses for the tree) / (Number of parentheses generated by the tree).

The authors also set up Martin's algorithmic algo- rithms and ran tests on the same number of sentences and our algorithms achvived better results. Specifically, when testing 30 Vietnamese sentences on 200 data sentence training, the authors' improved method for parsing results (accuracy ≈67.7%) was better than that of Jurafsky and Martin (degrees)Eaccuracy ≈62.2%).

V. CONCLUSION

This paper presents the development of an algorithm for extracting the CFG grammar, PCFG from VietTreebank and an improved parsing algorithm from Martin's PCYK parser algorithm. We have built a system of automatic extraction of rules for the PCFG grammar and a parser for the Vietnamese sentence based on the PCFG grammar. With good parser, VietTreebank can be re-expanded by automatically analyzing the syntax of annotated sentences from the Vietnamese language taken directly from the Internet. However, if the current parser results are not high enough, VietTreebank can expand by semi-automatic, for example using the parser to annotate automatically the sentence syntax, then the commentator can edit each sentence if necessary. This method provides accurate accreditation for VietTreebank and minimizes effort for commentators.

REFERENCES

1. Nguyễn Phương Thái và các cộng sự, Báo cáo kết quả sản phẩm SP 7.3-Kho ngữ liệu tiếng Việt có chú giải, Quyển 1, 2009, KC01/01, Dự án VLSP,2009.14
2. Nguyễn Quốc Thể, Lê Thanh Hương, Phân tích cú pháp tiếng Việt sử dụng văn phạm phi ngữ cảnh từ vựng hóa kết hợp xác suất, FAIR conference, Nha Trang, Việt Nam, 2007.
3. Uỷ ban khoa học xã hội Việt Nam, Ngữ pháp tiếng Việt, NXB Khoa học Xã hội, Hànội, 1993.
4. Chomsky, N. Three models for the description of language. IRI Transactions on Information Theory, 2(3), 113-124. 1956.
5. D.Jurafsky, J. H Martin, Introduction to natural
6. language processing, computational linguistics and speech recognition, Prentice Hall, Second Edition, 2009.
7. Nguyen P.T., Xuan L. V., Nguyen T. M. H., Nguyen V. H., Le H. P., Building a largesyntactically-annotated corpus of Vietnamese. In Proceedings of the 3rd Linguistic AnnotationWorkshop,ACL-IJCNLP, Singapore. 2009.
8. Phuong Le-Hong, Azim Roussanaly, Thi Minh Huyen Nguyen, Mathias Rossignol, An empirical study of maximum entropy approach for part-of-speech tagging of Vietnamese texts, TALN 2010, Montréal, 19–23 juillet 2010.
9. http://staff.science.uva.nl/~rvalenti/projects/lsp/PCFGReport.pdf.126.

AUTHORS PROFILE

Dr. Phan Thi Ha is currently a lecturer at the Faculty of Information Technology at Posts and Telecommunications Institute of Technology in Vietnam. She received a B.Sc.in Math & Informatics, a M.Sc. in Mathematic Guarantee for Computer Systems and a PhD. in Information Systems in 1994, 2000 and 2013, respectively. Her research interests include machine learning, natural language processing and mathematics applications. Email:hathiphan@yahoo.com and **hapt@ptit.edu.vn**

Ha Hai Nam, Associate Professor: Deputy Director of Reseach Institute of Posts and Telecommunications (RIPT), Area of scientific interests - modeling, control systems, algorithmization; machine learning; data mining, Email: namhh@ptit.edu.vn

Profile Dr. Do Xuan Cho is currently a lecturer at the Faculty of Information Technology at Posts and Telecommunications Institute of Technology in Vietnam**,** In 2008, received a bachelor's degree in the Saint Petersburg Electrotechnical University "LETI" on a specialty "Computer science and computer facilities", Russia. In 2010, graduated a master's from the Saint Petersburg Electrotechnical University "LETI" on a specialty "Computer science and computer facilities", Russia. In 2013, received a PhD in the Saint Petersburg Electrotechnical University "LETI", on a specialty CAD. Russia. Area of scientific interests - modeling, control systems, algorithmization, Email: chodx@ptit.edu.vn