Vietnamese Noun Phrase Chunking based on Conditional Random Fields

Nguyen Thi Huong Thao[†], Nguyen Phuong Thai[†], Nguyen Le Minh[‡], Ha Quang Thuy[†]
[†]*College of Technolgy, Vietnam National University, Hanoi*
{ thaonth, thainp, thuyhq}@vnu.edu.vn
[‡]*Japan Advanced Institute of Science and Technology*
nguyenml@jaist.ac.jp

Abstract

Noun phrase chunking is an important and useful task in many natural language processing applications. It is studied well for English, however with Vietnamese it is still an open problem. This paper presents a Vietnamese Noun Phrase chunking approach based on Conditional random fields (CRFs) models. We also describe a method to build Vietnamese corpus from a set of hand annotated sentences. For evaluation, we perform several experiments using different feature settings. Outcome results on our corpus show a high performance with the average of recall and precision 82.72% and 82.62% respectively.

1. Introduction

In recent years, applications in natural language processing such as text summarization, question answering and machine translation often require syntactic analysis at various levels including full parsing and chunking. The choice of which syntactic analysis level should be used depends on the specific priority of an application: speed or accuracy. The advantage of chunking in comparison with full parsing is the high speed. Since noun phrases take an important role is these applications, noun phrase chunking is also an important task. The importance of NP chunking derives from the fact that it is used in many applications such as information extraction, co-reference resolution, argument structure identification, etc.

A text chunker divides sentences into non-overlapping phrases. Specifically, NP chunking aims to identifying non-recursive noun phrases. This task was originally proposed by Stenven Abney [3]. The author's model divided a text into correlated phrases. Then, several other authors have been focused on low-level noun group identification, such as terminology extraction [follow 10]. However until Lance Ramshaw and Mitch Marcus proposed chunking method by using machine learning with good results 1995 [10], this task is known widely and inspired many others to study. The CONLL2000[1] share task was English text chunking. There are eleven systems applied in this conference. Kudoh and Matsumoto system based on support vector machines method achieved the best performance. The precision, recall and F1 of all chunks were 93.45%, 93.51% and 93.48% respectively; and 93.72%, 94.02% and 93.87% with NPs [17] A number of other approaches were applied recently, such as Conditional Random Fields (CRFs) [13, 15, 20, 22, 23], Maximum entropy markov models [16], combined systems (CRFs and SVMs) [9] also got high performance.

In general, English text chunking achieved good results. However, Vietnamese NP chunking has not been studied much yet due to the lack of Vietnamese language processing resources and tools. In this paper, we present an investigation of using CRFs, a powerful statistical learning method to perform Vietnamese NP chunking task. We first build a corpus extracted from a set of hand annotated sentences. Then we perform several experiments using various feature configurations. We also investigate the effects of using different sizes of training data. Experimental results on our corpus show effective of the model.

The rest of this paper is as follow: Section 2 describes several important characteristics of Vietnamese NP. Section 3 proposes a method to build Vietnamese NP corpus. Section 4 present CRFs models and section 5 introduces our model based on CRFs. Section 6 shows experimental results. Finally, we draw the conclusions and future work in section 7.

[1] http://www.cnts.ua.ac.be/conll2000/chunking/

978-0-7695-3846-4/09 $26.00 © 2009 IEEE
DOI 10.1109/KSE.2009.43

2. Vietnamese Noun Phrase Characteristics

Vietnamese is the official language of Vietnam. Many words in Vietnamese are borrowed from Chinese. Originally, it is written in Chinese-like writing system. The current writing system of Vietnamese is a modification of Latin alphabet, with additional diacritics for tones and certain letters. Vietnamese, like many languages in Southeast Asia, is an isolating language, which do not use morphological making of case, gender, number or tense. One word can be made of one or more syllables. Another important characteristic is one word can belong to many word classes such as noun, verb or adjective class. For example, "thắng lợi" (succeed) is made of two syllables "thắng" (win) and "lợi" (profit). If we consider the word meaning, "thắng lợi" belongs to verb class; however in other contexts, this word can be on alternative classes:

(1) ***Thắng lợi*** của chúng ta rất to lớn (Our success is very great)

(2) Chúng ta đang ***thắng lợi*** lớn (We are succeeding)

(3) Chúng ta rất ***thắng lợi*** trong việc này (We are very successful in this work)

"thắng lợi" in (1), (2), (3) is respectively a noun, a verb and an adjective. So, word class identification is mainly based on its surrounding context. Unlike English, one word can be derived from an existing word by adding prefix and suffix. Furthermore, in structure of Vietnamese NPs, head noun can receive features depicted by verbs, adjectives, numerals, nouns or pronouns, etc. So, Vietnamese NPs recognition comes up against more difficulties.

A Vietnamese noun phrase consists of a head noun, optionally accompanied by pre-modifiers and post-modifiers:

Table 1: Structure of Vietnamese noun phrase

Pre-modifiers			Head Noun
P_{-3}	P_{-2}	P_{-1}	Head Noun
Totality	Quantifier	Classifier	Noun

Head Noun	Post-modifiers	
Head Noun	P_{+1}	P_{+2}
Noun	Attributive modifiers	Demonstrative words

A pre-modifier can be located at three possible positions (Table 1). These positions are stable, and cannot be permuted each other. The number of possible cases is limited. Post-modifiers are more complicated than pre-modifiers. Many syntactic constituents can occur concurrently after the head noun. They can be nouns, verbs, adjectives, numerals, or pronouns. In addition, these substantive words can be combined to phrases such as noun phrases, verb phrases, adjective phrases, etc. to take the part of this position. As a result, the structure of post-modifiers is very complicated; A Vietnamese NP can contain other NPs. Several examples of Vietnamese NPs are presented below:

Tất cả *All*	sinh viên *students*	trường Đại học Công nghệ *College of Technogy*
Totality	Head Noun	Attributive modifiers

In this example, "Tất cả sinh viên trường Đại học Công nghệ" (All students of College of Technology) is a NP. "trường Đại học Công nghệ" is also a NP modifying the head noun.

Another example: "sự hoạch định chính sách ấy" (that policymaking) is a NP.

Sự (Noun)	hoạch định (Verb)	chính sách (Noun)	ấy (Pronoun)
Head Noun	Attributive modifiers		Demonstrative words

Next part will introduce the method to build Vietnamese corpus.

3. Corpus Construction

Our corpus is derived automatically from Viet Treebank [12], a corpus consisting of 5329 hand annotated sentences[2]. In Viet Treebank, there are three annotation levels including word segmentation, part-of-speech (POS) tagging, and syntactic labeling. Word segmentation identifies word boundary in sentences. POS tagging assigns correct POS tags to words. Syntactic labeling recognizes constituency tags, functional tags, and null-element tags. For English, base NPs are noun phrases without post-modifiers. Ramshaw and Marcus [10] identify base NPs as the initial portions of non-recursive NP up to the head. However, if we apply the English definition of base NP for Vietnamese, it is too narrow (since in Vietnamese, modifiers which are content words – or phrases – are all post-modifiers). Therefore, we extract NPs following the structure described in Section 2, including pre-modifiers, head noun and post-modifiers except complex post-modifiers such as prepositional phrases and clauses. We proposed several rules to extract NPs depending on the depth of constituent tree. We present several examples in Figure 1 and Figure 2. Because one Vietnamese word can be composed of one

[2] At the moment, Viet Treebank consists of nearly 10000 sentences [12], but we have not updated yet because of time restriction,

or more syllables, we use underlines to link syllables in a word. For instances, "cuộc đời" becomes cuộc_đời, "xinh đẹp" becomes xinh_đẹp.

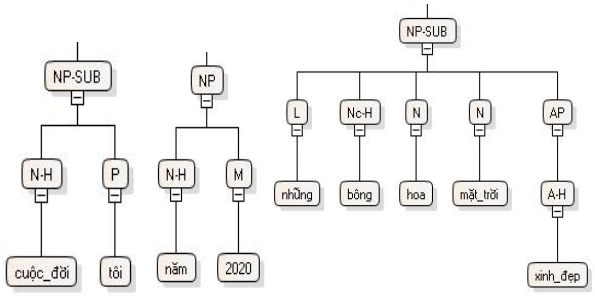


Figure 1: Examples of Vietnamese NP

Two first examples in Fingure 1: "cuộc đời tôi" (my life) and "năm 2020" (the year 2020), the head noun is modified by a pronoun and a number respectively. The depth of the NP constituent is 1.

The NP in third example "những bông hoa mặt trời xinh đẹp" (beautiful sunflowers) is more complicated. In this example, the head noun is modified by both pre-modifiers and post-modifiers. The pre-modifier is a quantifier; And post-modifiers inlude two nouns and an adjecive phrase. The depth of the NP constituent is 2.

In Figure 2, the NP in the first example: "Bộ trưởng Bộ Tài nguyên & môi trường" (Minister of the ministry of natural resources and environment) is a NP where "Bộ trưởng" is the head noun which is modified attribution by a NP "Bộ tài nguyên & môi trường". This is a recursive NP including two NPs inside. The second example, "cơ sở khám chữa bệnh" (the health clinic) is a NP, where "khám chữa bệnh" (examine and treat medically) is a verb phrase modifying attribution to the head noun "cơ sở" (place). NP constituents of two these examples is 3 in depth.

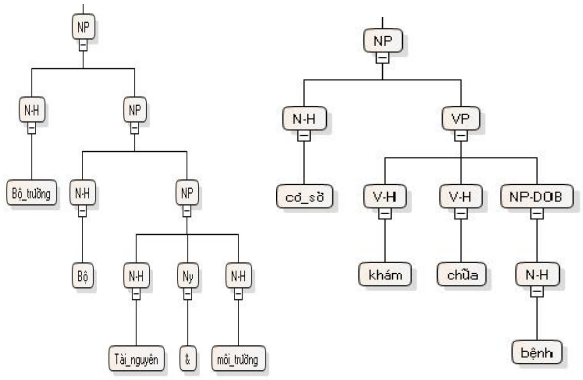


Figure 2: Examples of Vietnamese NP

From several examples above, we can see the diversity of Vietnamese NP structures, especially post-modifiers. Based on the structure of NP constituents, we select NPs satisfying following criteria:

- The depth of the NP constituent is 1.
- The post-modifier that the depth of its constituent is 1 is not a prepositional phrase.
- The post-modifier is a NP or VP constituent 2 in depth.
- If the depth of NP's branch is greater than 3, we select only initial portions NP up to the head.

With respect to NPs including conjunction "và" (and), the NP phrase can be considered as a single NP spanning the conjunction or separate NPs depending on the structure of NP constituent. Figure 3 give examples of these cases.

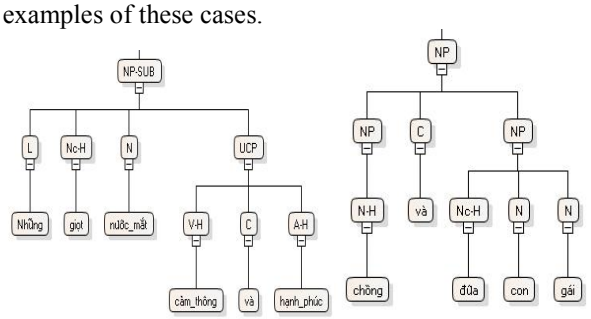


Figure 3: Examples of Vietnamese NP including conjunction

The first phrase "những giọt nước mắt cảm thông và hạnh phúc" (sympathetic and happy tears) is considered as a NP. However, "chồng và đứa con gái" (husband and daughter) is separated into two NPs.

Other special cases such as NPs containing double quotation marks, hyphen, etc. we also built suitable rules. However, due to the diversity of NP structure, these rules may not cover all cases. So, after this process, we review again and correct error manually.

4. Conditional Random Fields

Conditional Random fields was originally introduced by Lafferty [11], is a statistical sequence modeling framework for labeling and segmenting sequential data. Overcoming weakness of HMM and MEMM, CRFs is appreciated as one the best methods for labeling tasks.

CRFs calculate conditional probability distributions p(y|x) of label sequence $\mathbf{y} = (y_1, ..., y_n) \in \Upsilon^n$ given variable sequence $\mathbf{x} = (x_1, ..., x_n) \in \aleph^n$:

$$P(\mathbf{y} \mid \mathbf{x}) = \frac{1}{Z(\mathbf{x})} \exp\left(\sum_i \sum_k \lambda_k t_k(\mathbf{y}_{i-1}, \mathbf{y}_i, \mathbf{x}) + \sum_i \sum_k \mu_k s_k(\mathbf{y}_i, \mathbf{x}) \right)$$

Where Z(x) is normalization factor to ensure a proper probability:

$$Z(\mathbf{x}) = \sum_{y} \exp\left(\sum_{i} \sum_{k} \lambda_k t_k(\mathbf{y}_{i-1}, \mathbf{y}_i, \mathbf{x}) + \sum_{i} \sum_{k} \mu_k s_k(\mathbf{y}_i, \mathbf{x}) \right)$$

And t_k is a transformation feature of entire observation sequence x from y_{i-1} state to y_i state; s_k is a state feature of observation sequence at state y_i. For example:

$$t_k = \begin{cases} 1 \text{ if } x_{i-1} = \text{“tất cả”}, x_i = \text{“sinh viên”}, y_{i-1} = B, y_i = I \\ 0 \text{ otherwise} \end{cases}$$

$$s_k = \begin{cases} 1 \text{ if } x_i = \text{“tất cả” and } y_i = I \\ 0 \text{ otherwise} \end{cases}$$

λ_k and μ_k are parameters estimated from training data. To train CRFs given training data, several advanced convex optimization techniques is commonly used to maximize the likelihood such as L-BFGS, Newton, etc.

CRFs has been applied in many natural language processing applications and achieved high performance. In chunking task, CRFs has been used in many systems of different languages, such as English, Chinese, Hebrew, Korean, Indian languages [13, 15, 20, 22, 23, 24], etc. and becomes one of the best methods to identify chunks.

5. Vietnamese NP Chunking Model

We can treat noun phrase chunking as the tagging problem. Assume $\mathbf{x} = (x_1, ..., x_n)$ is the input sentence, consists of *n* words, we must determine sequence of tag $\mathbf{y} = (y_1, ..., y_n)$ We used tag set {B, I, O} where *B* denotes the beginning of a NP; I denotes inside of a NP; And O denote outside of a NP.

This is IOB2 data presentation model that was originally introduced by Ramshaw and Marcus [10]. NPs are extracted by identifying the beginning and the end of NPs. Besides, there are several different methods to present data, such as IOB1, IOE1, IOE2, etc. In this paper, we use only IOB2 format in all experiments. An instance of IOB2 format as follow:

những	L	B-NP
bông	Nc-H	I-NP
hoa	N	I-NP
mặt_trời	N	I-NP
xinh_đẹp	A-H	I-NP
ngả	V-H	O
bóng	N-H	B-NP
xuống	E-H	O
…		

The two first columns are lexical and POS information, the third column is IOB2 tag.

We applied CRFs to our system. The frame of Vietnamese NP chunker is described in Figure 4:

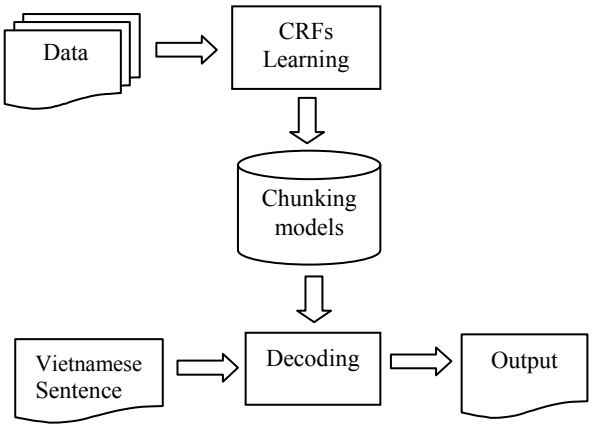


Figure 4: Vietnamese NP chunking system

6. Experiments

Our experiments with CRFs were conducted using CRF++[3] toolkit – a C/C++ implementation of CRFs for labeling and segmenting sequence data. Two types of features: unigram features and bigram features are used. We use standard measures: accuracy (at tag level); Precision, Recall and F1 (at NP level) to evaluate the performance of our chunking system.

For measuring the performance of each experiment, we use the Perl script conlleval[4] provided by CoNLL-2000. Several experiments are conducted as follows:

a. Effect of Feature Set

Performance of CRFs-based NP chunker depends on quality of feature set. Especially, Vietnamese NPs are complex; NPs identification depends on appearance context of surrounding current words.

The test set and train set are chosen randomly according to the scale of 1:2 – the common partition in large corpus. Table 2 listed details on our corpus used in this study:

Table 2: Statistics of the corpus

	Number of sentences	Number of NPs	Number of tokens
Train set	3552	78751	18165
Test set	1777	39005	9136
Total	5329	117756	27301

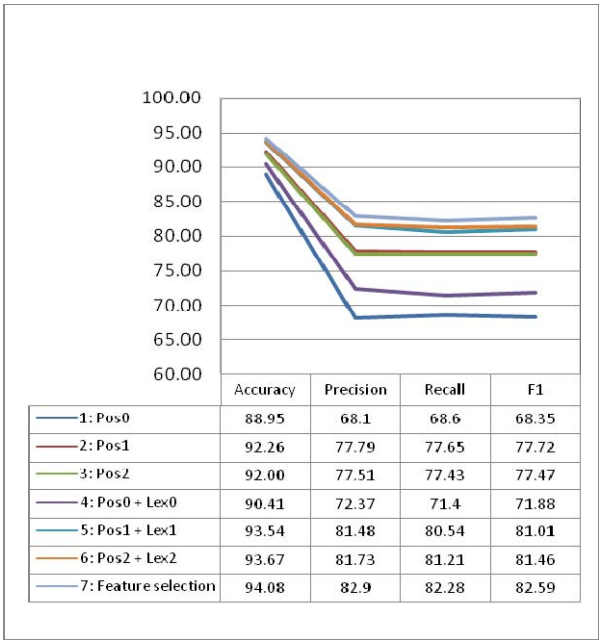
[3] http://crfpp.sourceforge.net/

[4] http://www.cnts.ua.ac.be/conll2000/chunking/output.html

We utilize POS and lexical information as the features. Denote Pos0 and Lex0 are POS and lexical information at current position. Pos_n and Lex_n are POS and lexical information in n window where n is window size. Consider the instance in section 5, assume that the current token is "mặt_trời", we have:

L_0	: "mặt_trời"	P_0	: N
L_1	: "xinh_đẹp"	P_1	: A-H
L_{-1}	: "hoa"	P_{-1}	: N

Three first experiments used only POS information with window size 0, 1, 2. After that, lexical information is added. We also use combination features (POS and lexical) in these experiments. Feature selection is made in experiment 7. Figure 5 illustrates output results.



	Accuracy	Precision	Recall	F1
1: Pos0	88.95	68.1	68.6	68.35
2: Pos1	92.26	77.79	77.65	77.72
3: Pos2	92.00	77.51	77.43	77.47
4: Pos0 + Lex0	90.41	72.37	71.4	71.88
5: Pos1 + Lex1	93.54	81.48	80.54	81.01
6: Pos2 + Lex2	93.67	81.73	81.21	81.46
7: Feature selection	94.08	82.9	82.28	82.59

Figure 5: Effect of feature set to performance of CRF-based NP chunker

Table 3: The feature set of experiment 7

Unigram feature	
Lexical	L_{-2}, L_{-1}, L_0, L_1, L_2, $L_{-2}L_{-1}$, $L_{-1}L_0$, L_0L_1, L_1L_2
POS	P_{-3}, P_{-2}, P_{-1}, P_0, P_1, P_2, P_3, $P_{-2}P_{-1}$, $P_{-1}P_0$, P_0P_1, P_1P_2, $P_{-2}P_{-1}P_0$, $P_{-1}P_0P_1$, $P_0P_1P_2$
Combination	$L_{-2}P_{-2}$, $L_{-1}P_{-1}$, L_0P_0, L_1P_1, L_2P_2, $P_0P_1L_0$, $P_{-1}P_0L_0$
Bigram feature	
Lexical	L_0, L_{-1}, $L_{-1}L_0$
POS	P_{-2}, P_{-1}, P_0, P_1, P_0P_1, $P_{-1}P_0$
Combination	$L_{-1}P_{-1}$, L_0P_0

From the results, we see that part-of-speech and lexical information current word to left and right impacts to performance. If we use only POS information, POS2 is little worse than POS1. But, adding lexical information brings better results in all cases. F1 in experiment 5 is better 3.29% than experiment 2. Expanding window size to 1 achieves 9.13% better than using only current word and POS. The best performance achieved F1 = 82.59% in experiment 7 when we take feature selection. The feature set in experiment 7 is detailed in Table 3.

Table 4: Performance of CRFs-base NP chunker

Case	Num of NPs	Acc	Pre	Re	F1
1	9136	94.08	82.90	82.28	82.59
2	9113	93.77	82.28	82.21	82.24
3	9130	93.88	82.53	82.50	82.52
4	**8949**	**94.30**	**83.27**	**83.28**	**83.28**
5	9173	94.15	82.64	82.85	82.74
Average	**9100**	**94.04**	**82.72**	**82.62**	**82.67**

Using these features, we perform 5 times choosing randomly train set and test set. The outcome is shown in Table 4. The fourth case archives the best performance, but the differences among five cases are not considerable. The average of F1 is 82.67%.

b. Effect of Corpus Size

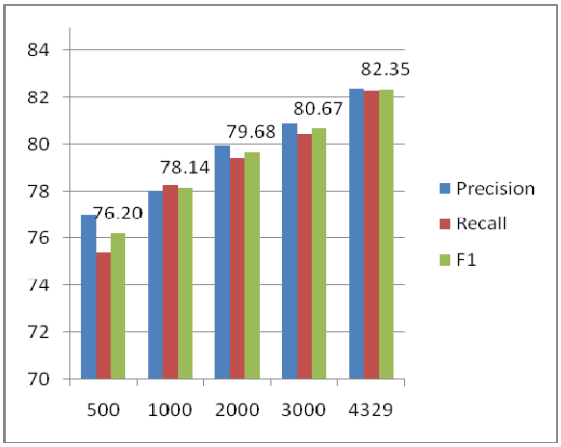


Figure 6: Effect of training size to performance of CRF-based NP chunker

To investigate the effect of the size on the training set, we pick randomly different sizes of training sets, including 500, 1000, 2000, 3000, 4329 sentences. The test set is fixed 1000 sentences. The feature set is used as in experiment 7 of previous section. Figure 6 presents obtained results where numbers are F1

measures of each experiment. From this figure, we see that when increasing training set, we can get better performance.

b. Error Analysis and Discussion

From the obtained output, we detect several cases predicted incorrectly. For example, the NP "nghề nuôi tôm sú" (prawn-farming) is identified as follow:

nghề	N-H	B-NP	B-NP
nuôi	V-H	I-NP	O
tôm_sú	N-H	I-NP	B-NP

The last column is the predicted tag. In this example, our chunker divided "nghề nuôi tôm sú" into two NPs: "nghề" (industry) and "tôm_sú" (prawn). Note that, in our corpus, "nuôi" (nourish) is a verb that is outside chunks in most cases.

Another example: "đại diện Viện kiểm sát" (the representative of people's procurancy):

đại_diện	N-H	B-NP	B-NP
Viện_Kiểm_sát	Np-H	I-NP	B-NP

In this example, the chunker also divided the NP into two NPs. The reason may be is the POS information (Np-H) of "Viện_Kiểm_sát". In many examples, words having Np-H POS information are often the beginning of a NP. Similar to this example, the NP "lọ thuộc Pennicillin" (the Penicillin phial) is predicted into two NPs:

lọ	N-H	B-NP	B-NP
thuốc	N	I-NP	I-NP
Penicillin	Np-H	I-NP	B-NP

However, many recursive NPs are divided well.

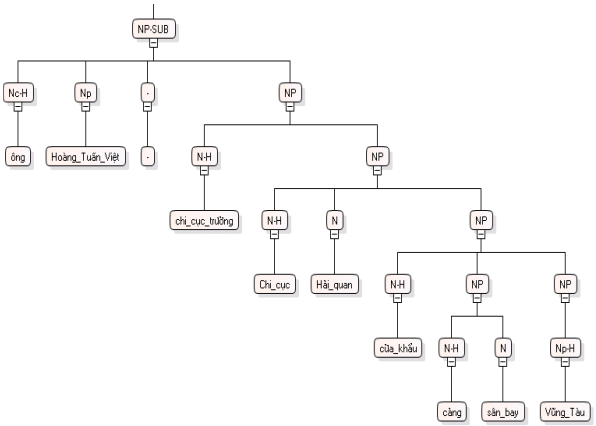


Figure 4: An example of recursive Vietnamese NP

For example, "Ông Hoàng Tuấn Việt – chi cục trưởng Chi cục Hải quan cửa khẩu cảng sân bay Vũng Tàu" (Mr.Hoang Tuan Viet – branch manager of the Customs Department of Vung Tau Airport's harbour) is a recursive NP (Figure 4). The chunker divided correctly this phrase into three NPs:

ông	Nc-H	B-NP	B-NP
Hoàng_Tuấn_Việt	Np	I-NP	I-NP
-	-	O	O
chi_cục_trưởng	N-H	B-NP	B-NP
Chi_cục	N-H	B-NP	I-NP
Hải_quan	N	I-NP	I-NP
cửa_khẩu	N-H	B-NP	B-NP
cảng	N-H	I-NP	I-NP
sân_bay	N	I-NP	I-NP
Vũng_Tàu	Np-H	I-NP	I-NP

The results above show that CRFs-based learning is a potential approach to solve Vietnamese NP chunking task. With suitable feature set and large enough of training data, our system brings promising performance.

7. Conclusion and Future Work

Our work concentrated on solving Vietnamese noun phrase chunking problem. First, we have showed that Vietnamese NPs identification meet with difficulties because of complicated characteristics of Vietnamese NPs. Then, we have introduced a method to construct Vietnamese NP chunking corpus from Viet treebank. Performing several experiments based on CRFs models, the experimental results have shown the efficiency of our approach. In all experiments, we used only features related to part-of-speech and lexical information. Our future works will concentrate to the effects of data presentation methods and some different features; also, we will apply some other methods such as support vector machines, combined system for comparison. This work only deals with Vietnamese NP identification. Other kinds of chunks will be also studied in near future.

Our chunking system will be soon released for research purposes, and we believe that it would be helpful for the Vietnamese natural language processing community.

Acknowledgements. This paper is supported by a national project named Building Basic Resources and Tools for Vietnamese Language and Speech Processing, KC01.01/06-10.

References

[1] Diệp Quang Ban, Hoàng Bân, "Vietnamese Grammar" *Education Publishing House*, HaNoi, 2004

[2] Nguyễn Chí Hoà. "Practical Vietnamese Grammar". *Hanoi National University Publishing House*, 2004

[3] Abney, Steven. 1991, "Parsing by chunks". In Berwick, Abney, and Tenny, editors, Principle-Based Parsing. Kluwer Academic Publishers

[4] Andrew McCallum, Freitag, and Pereia, "Maximum entropy markov models for information extraction and segmentation", *In Proc. International Conference on Machine Learning*, 2000

[5] Andrew McCallum, "Efficiently Inducing Features of Conditional Random Fields", *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2003

[6] Akshar Bharathi, Prashanth R.Mannem, "Introduction to the Shallow Parsing Contest for South Asia Languages", *Proceedings of the IJCAI-2007 Workshop on Shallow Parsing for South Asian languages*, 2007

[7] Eric Brill, "A Corpus-Based Approach to Language Learning", *phD thesis*, *University of Pennsylvania*, 1993

[8] Erik F. Tjong Kim Sang, Sabine Buchholz, "Introduction to the CoNLL-2000 Shared Task: Chunking", *Proceedings of CoNLL-2000 and LLL-2000*, pages 127-132, Lisbon, Portugal, 2000

[9] Fang Xu, Chengqing Zong, "A Hybrid Approach to Chinese Base Noun Phrase Chunking", *Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing*, pages 87–93, Sydney, July 2006

[10] Lance A.Ramshaw, Mitchell P.Marcus, "Text Chunking using Transformation-Based Learning", *Proceedings of the ACL 3rd Workshop on Very Large Corpora*, 1995, pp. 82-94

[11] Lafferty, John D., McCallum, Andrew, Pereira, Fernando C. N., "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data", *Proceedings of the Eighteenth International Conference on Machine Learning (ICML 2001)*, Morgan Kaufmann Publishers, 2001, pp. 282

[12] Nguyen Phuong Thai, Vu Xuan Luong, Nguyen Thi Minh Huyen, Nguyen Van Hiep, Le Hong Phuong. Building a Large Syntactically-Annotated Corpus of Vietnamese. *Proceedings of the 3rd Linguistic Annotation Workshop (LAW) at ACL-IJCNLP 2009 (to appear)*.

[13] H.X. Phan, M.L. Nguyen, Y. Inoguchi, and S. Horiguchi. "High-Performance Training Conditional Random Fields for Large-Scale Applications of Labeling Sequence Data", *IEICE Transactions on Information and Systems*, Vol.E90-D, No.1, pp.13-21, 2007

[14] Rie Kubota Ando, Tong Zhang. "A High-Performance Semi-Supervised Learning Method for Text Chunking". *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, p.1-9, June 25-30, 2005, Ann Arbor, Michigan

[15] Sha, F., Pereira, F. "Shallow parsing with conditional random fields". *Technical Report MS-CIS-02-35*, University of Pennsylvania (2003)

[16] Sutton, C., McCallum, A. "An Introduction to Conditional Random Fields for Relational Learning". *In Introduction to Statistical Relational Learning*. Edited by Lise Getoor and Ben Taskar. MIT Press. (2006)

[17] Sun, Hunag, Wang, Xu. "Chinese Chunking Based on Maximum Entropy Markov Models", *International Journal of Computational Linguistics and Chinese Language Processing,* p115-136, 2006

[18] Taku Kudo, Yuji Matsumoto. "Use of Support Vector Learning for Chunk Identification". *Proceedings of CoNLL-2000 and LLL-2000*, pages 142-144, Lisbon, Portugal, 2000

[19] Taku Kudo, Yuji Matsumoto. "Chunking with Support Vector Machines", *Proceedings of the NAACL 2001*, pages 192-199

[20] Wenliang Chen, Yujie Zhang, Hitoshi Isahara. "Chinese Chunking based on Conditional Random Fields". *NLP2006,* Yokohama, Japan, pp. 149-152, Mar. 2006

[21] Yoav Goldberg, Meni Adler, Michael Elhadad. "Noun Phrase Chunking in Hebrew Influence of Lexical and Morphological Features". *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the ACL*, p.689-696, July 17-18, 2006, Sydney, Australia

[22] Yong-Hun Lee, Mi-Young Kim, and Jong-Hyeok Lee. "Chunking Using Conditional Random Fields in Korean Texts". *Lecture Notes in Artificial Intelligence IJCNLP* 2005

[23] Yongmei Tan, Tianshun Yao, Qing Chen and Jingbo Zhu. "Applying Conditional Random Fields to Chinese Shallow Parsing". *The 6th International Conference on Intelligent Text Processing and Computational Linguistics* (CICLing-2005)

[24] Wengliang Chen, Yujie Zhang, Hotoshi Isahara. "An Empirical Study of Chinese chunking" *In Proceedings of the 44th Annual Meeting of ACL*, pages 97-104