An Empirical Study of Vietnamese Noun Phrase Chunking with Discriminative Sequence Models

Le Minh Nguyen **Huong Thao Nguyen** and **Phuong Thai Nguyen**
School of Information Science, JAIST College of Technology, VNU
nguyenml@jaist.ac.jp {thaonth, thainp}@vnu.edu.vn

Tu Bao Ho and **Akira Shimazu**
Japan Advanced Institute of Science and Technology
{bao,shimazu}@jaist.ac.jp

Abstract

This paper presents an empirical work for Vietnamese NP chunking task. We show how to build an annotation corpus of NP chunking and how discriminative sequence models are trained using the corpus. Experiment results using 5 fold cross validation test show that discriminative sequence learning are well suitable for Vietnamese chunking. In addition, by empirical experiments we show that the part of speech information contribute significantly to the performance of there learning models.

1 Introduction

Many Natural Language Processing applications (i.e machine translation) require syntactic information and tools for syntactic analysis. However, these linguistic resources are only available for some languages(i.e English, Japanese, Chines). In the case of Vietnamese, currently most researchers have focused on word segmentation and part of speech tagging. For example, Nghiem et al (Nghiem, Dinh, Nguyen, 2008) has developed a Vietnamese POS tagging. Tu (Tu, Phan, Nguyen, Ha, 2006) (Nguyen, Romary, Rossignol, Vu, 2006)(Dien, Thuy, 2006) have developed Vietnamese word segmentation.

The processing of building tools and annotated data for other fundamental tasks such as chunking and syntactic parsing are currently developed. This can be viewed as a bottleneck for developing NLP applications that require a deeper understanding of the language. The requirement of developing such tools motives us to develop a Vietnamese chunking tool. For this goal, we have been looking for an annotation corpus for conducting a Vietnamese chunking using machine learning methods. Unfortunately, at the moment, there is still no common standard annotated corpus for evaluation and comparison regarding Vietnamese chunking.

In this paper, we aim at discussing on how we can build annotated data for Vietnamese text chunking and how to apply discriminative sequence learning for Vietnamese text chunking. We choose discriminative sequence models for Vietnamese text chunking because they have shown very suitable methods for several languages(i.e English, Japanese, Chinese) (Sha and Pereira, 2005)(Chen, Zhang, and Ishihara, 2006) (Kudo and Matsumoto, 2001). These presentative discriminative models which we choose for conducting empirical experiments including: Conditional Random Fields (Lafferty, McCallum, and Pereira, 2001), Support Vector Machine (Vapnik, 1995) and Online Prediction (Crammer et al, 2006). In other words, because Noun Phrase chunks appear most frequently in sentences. So, in this paper we focus mainly on empirical experiments for the tasks of Vietnamese NP chunking.

We plan to answer several major questions by using empirical experiments as follows.

- Whether or not the discriminative learning models are suitable for Vietnamese chunking problem?
- We want to know the difference of SVM, Online Learning, and Conditional Random Fields for Vietnamese chunking task.
- Which features are suitable for discriminative learning models and how they contribute to the performance of Vietnamese text chunking?

The rest of this paper is organized as follows: Section 2 describes Vietnamese text chunking with discriminative sequence learning models. Section 3 shows experimental results and Section 4 dis-

Proceedings of the 7th Workshop on Asian Language Resources, ACL-IJCNLP 2009, pages 9–16,
Suntec, Singapore, 6-7 August 2009. ©2009 ACL and AFNLP

cusses the advantage of our method and describes future work.

2 Vietnamese NP Chunking with Discriminative Sequence Learning

Noun Phrase chunking is considered as the task of grouping a consecutive sequence of words into a NP chunk lablel. For example: “[NP Anh Ay (He)] [VP thich(likes)] [NP mot chiec oto(a car)] ”

Before describing NP chunking tasks, we summarize the characteristic of Vietnamese language and the background of Conditional Random Fields, Support Vector Machine, and Online Learning. Then, we present how to build the annotated corpus for the NP chunking task.

2.1 The characteristic of Vietnamese Words

Vietnamese syllables are elementary units that have one way of pronunciation. In documents, they are usually delimited by white-space. Being the elementary units, Vietnamese syllables are not undivided elements but a structure. Generally, each Vietnamese syllable has all five parts: first consonant, secondary vowel, main vowel, last consonant and a tone mark. For instance, the syllable tu.n (week) has a tone mark (grave accent), a first consonant (t), a secondary vowel (u), a main vowel () and a last consonant (n). However, except for main vowel that is required for all syllables, the other parts may be not present in some cases. For example, the syllable anh (brother) has no tone mark, no secondary vowel and no first consonant. In other case, the syllable hoa (flower) has a secondary vowel (o) but no last consonant.

Words in Vietnamese are made of one or more syllables which are combined in different ways. Based on the way of constructing words from syllables, we can classify them into three categories: single words, complex words and reduplicative words (Mai,Vu, Hoang, 1997).

The past of speechs (Pos) of each word in Vietnamese are mainly sketched as follows.

A Noun Phrase (NP) in Vietnamese consists of three main parts as follows: the noun center, the prefix part, and the post fix part. The prefix and postfix are used to support the meaning of the NP. For example in the NP ”ba sinh vien nay”, the noun center is ”sinh vien”, and the prefix is ”ba (three)”, the postfix is ”nay”.

Vietnamese Tag	Equivalent to English Tag
CC	Coordinating conjunction)
CD	Cardinal number)
DT	Determiner)
V	Verb
P	Preposition
A	Adjective
LS	List item marker
MD	Modal
N	Noun

Table 1: Part of Speeches in Vietnamese

2.2 The Corpus

We have collected more than 9,000 sentences from several web-sites through the internet. After that, we then applied the segmentation tool (Tu, Phan, Nguyen, Ha, 2006) to segment each sentences into a sequence of tokens. Each sequence of tokens are then represented using the format of CONLL 2000. The details are sketched as follows.

Each line in the annotated data consists of three columns: the token (a word or a punctuation mark), the part-of-speech tag of the token, and the phrase type label (label for short) of the token. The label of each token indicates whether the token is outside a phrase (O), starts a phrase (B-⟨PhraseType⟩), or continues a phrase (I-⟨PhraseType⟩).

In order to save time for building annotated data, we made a set of simple rules for automatically generating the chunking data as follows. If a word is not a ”noun”, ”adjective”, or ”article” it should be assigned the label ”O”. The consecutive words are NP if they is one of type as follows: ”noun noun”; ”article noun”, ”article noun adjective”. After generating such as data, we ask an expert about Vietnamese linguistic to correct the data. Finally, we got more than 9,000 sentences which are annotated with NP chunking labels.

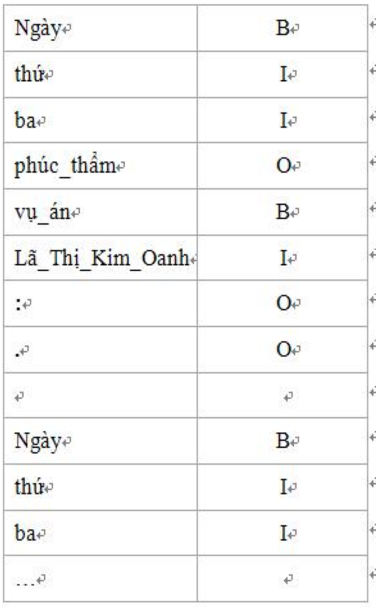
Figure 1 shows an example of the Vietnamese chunking corpus.

2.3 Discriminative Sequence Learning

In this section, we briefly introduce three discriminative sequence learning models for chunking problems.

2.3.1 Conditional Random Fields

Conditional Random Fields (CRFs) (Lafferty, McCallum, and Pereira, 2001) are undirected graphical models used to calculate the conditional



Ngày	B
thứ	I
ba	I
phúc_thẩm	O
vụ_án	B
Lã_Thị_Kim_Oanh	I
:	O
.	O
Ngày	B
thứ	I
ba	I
...	

Figure 1: An Example of the Vietnamese chunking corpus

probability of values on designated output nodes, given values assigned to other designated input nodes for data sequences. CRFs make a first-order Markov independence assumption among output nodes, and thus correspond to finite state machine (FSMs).

Let **o** = $(o_1, o_2, \ldots, o_T)$ be some observed input data sequence, such as a sequence of words in a text (values on T input nodes of the graphical model). Let **S** be a finite set of FSM states, each is associated with a label l such as a clause start position. Let **s** = $(s_1, s_2, \ldots, s_T)$ be some sequences of states (values on T output nodes). CRFs define the conditional probability of a state sequence given an input sequence to be

$$P_\Lambda(s|o) = \frac{1}{Z_o} exp\left(\sum_{t=1}^{T} F(s,o,t)\right) \quad (1)$$

where $Z_o = \sum_s exp\left(\sum_{t=1}^{T} F(s,o,t)\right)$ is a normalization factor over all state sequences. We denote δ to be the Kronecker-δ. Let $F(s,o,t)$ be the sum of CRFs features at time position t:

$$\sum_i \lambda_i f_i(s_{t-1}, s_t, t) + \sum_j \lambda_j g_j(o, s_t, t) \quad (2)$$

where $f_i(s_{t-1}, s_t, t) = \delta(s_{t-1}, l^{'})\delta(s_t, l)$ is a *transition* feature function which represents sequential dependencies by combining the label $l^{'}$ of the previous state s_{t-1} and the label l of the current state s_t, such as the previous label $l^{'}$ = AV (adverb) and the current label l = JJ (adjective). $g_j(o, s_t, t) = \delta(s_t, l)x_k(o,t)$ is a *per-state* feature function which combines the label l of current state s_t and a context predicate, i.e., the binary function $x_k(o,t)$ that captures a particular property of the observation sequence o at time position t. For instance, the current label is JJ and the current word is *“conditional“*.

Training CRFs is commonly performed by maximizing the likelihood function with respect to the training data using advanced convex optimization techniques like L-BFGS. Recently, there are several works apply Stochastic Gradient Descent (SGD) for training CRFs models. SGD has been historically associated with back-propagation algorithms in multilayer neural networks.

And inference in CRFs, i.e., searching the most likely output label sequence of an input observation sequence, can be done using Viterbi algorithm.

2.3.2 Support Vector Machines

Support vector machine (SVM)(Vapnik, 1995) is a technique of machine learning based on statistical learning theory. The main idea behind this method can be summarized as follows. Suppose that we are given l training examples (x_i, y_i), $(1 \leq i \leq l)$, where x_i is a feature vector in n dimensional feature space, and y_i is the class label {-1, +1 } of x_i.

SVM finds a hyperplane $w.x+b=0$ which correctly separates training examples and has maximum margin which is the distance between two hyperplanes $w \cdot x + b \geq 1$ and $w \cdot x + b \leq -1$. Finally, the optimal hyperplane is formulated as follows:

$$f(x) = \text{sign}\left(\sum_{1}^{l} \alpha_i y_i K(x_i, x) + b\right) \quad (3)$$

where α_i is the Lagrange multiple, and $K(x', x'')$ is called a kernel function, which calculates similarity between two arguments x' and x''. For instance, the Polynomial kernel function is formulated as follows:

$$K(x', x'') = (x' \cdot x'')^p \quad (4)$$

SVMs estimate the label of an unknown example x whether the sign of $f(x)$ is positive or not.

Basically, SVMs are binary classifier, thus we must extend SVMs to multi-class classifier in order to classify three or more classes. The pairwise classifier is one of the most popular methods to extend the binary classification task to that of K classes. Though, we leave the details to (Kudo and Matsumoto, 2001), the idea of pairwise classification is to build K.(K-1)/2 classifiers considering all pairs of classes, and final decision is given by their weighted voting. The implementation of Vietnamese text chunking is based on Yamcha (V0.33)[1].

2.3.3 Online Passive-Aggressive Learning

Online Passive-Aggressive Learning (PA) was proposed by Crammer (Crammer et al, 2006) as an alternative learning algorithm to the maximize margin algorithm. The Perceptron style for natural language processing problems as initially proposed by (Collins, 2002) can provide to state of the art results on various domains including text segmentation, syntactic parsing, and dependency parsing. The main drawback of the Perceptron style algorithm is that it does not have a mechanism for attaining the maximize margin of the training data. It may be difficult to obtain high accuracy in dealing with hard learning data. The online algorithm for chunking parsing in which we can attain the maximize margin of the training data without using an optimization technique. It is thus much faster and easier to implement. The details of PA algorithm for chunking parsing are presented as follows.

Assume that we are given a set of sentences x_i and their chunks y_i where $i = 1, ..., n$. Let the feature mapping between a sentence x and a sequence of chunk labels y be: $\Phi(x, y) = \Phi_1(x, y), \Phi_2(x, y), ..., \Phi_d(x, y)$ where each feature mapping Φ_j maps (x, y) to a real value. We assume that each feature $\Phi(x, y)$ is associated with a weight value. The goal of PA learning for chunking parsing is to obtain a parameter w that minimizes the hinge-loss function and the margin of learning data.

Algorithm 1 shows briefly the Online Learning for chunking problem. The detail about this algorithm can be referred to the work of (Crammer et al, 2006). In Line 7, the argmax value is computed by using the Viterbi algorithm which is similar to the one described in (Collins, 2002). Algorithm 1 is terminated after T round.

```
1  Input: $S = (x_i; y_i), i = 1, 2, ..., n$ in which
   $x_i$ is the sentence and $y_i$ is a sequence of
   chunks
2  Aggressive parameter $C$
3  Output: the model
4  Initialize: $w_1 = (0, 0, ..., 0)$
5  for t=1, 2... do
6    Receive an sentence $x_t$
7    Predict $y_t^* = \arg\max_{y \in Y}(w_t.\Phi(x_t, y_t))$
     Suffer loss: $l_t =$
     $w_t.\Phi(x_t, y_t^*) - w_t.\Phi(x_t, y_t) + \sqrt{\rho(y_t, y_t^*)}$
8    Set:$\tau_t = \frac{l_t}{||\Phi(x_t, y_t^*) - \Phi(x_t, y_t)||^2}$
9    Update:
     $w_{t+1} = w_t + \tau_t(\Phi(x_t, y_t) - \Phi(x_t, y_t^*))$
10 end
```

Algorithm 1: The Passive-Aggressive algorithm for NP chunking.

2.3.4 Feature Set

Feature set is designed through features template which is shown in Table 2. All edge features obey the first-order Markov dependency that the label (l) of the current state depends on the label (l') of the previous state (e.g., "l = I-NP" and "l' = B-NP"). Each observation feature expresses how much influence a statistic ($x(\mathbf{o}, i)$) observed surrounding the current position i has on the label (l) of the current state. A statistic captures a particular property of the observation sequence. For instance, the observation feature "l = I-NP" and "word$_{-1}$ is *the*" indicates that the label of the current state should be I-NP (i.e., continue a noun phrase) if the previous word is *the*. Table 2 describes both edge and observation feature templates. Statistics for observation features are identities of words, POS tags surrounding the current position, such as words and POS tags at -2, -1, 1, 2.

We also employ 2-order conjunctions of the current word with the previous ($w_{-1}w_0$) or the next word (w_0w_1), and 2-order and 3-order conjunctions of two or three consecutive POS tags within the current window to make use of the mutual dependencies among singleton properties. With the feature templates shown in Table 2 and the feature rare threshold of 1 (i.e., only features with occurrence frequency larger than 1 are included into the discriminative models)

[1] Yamcha is available at http://chasen.org/ taku/software/yamcha/

Edge feature templates	
Current state: s_i	Previous state: s_{i-1}
l	l'

Observation feature templates	
Current state: s_i	Statistic (or context predicate) templates: $x(\mathbf{o}, i)$
l	w_{-2}; w_{-1}; w_0; w_1; w_2; $w_{-1}w_0$; w_0w_1; t_{-2}; t_{-1}; t_0; t_1; t_2; $t_{-2}t_{-1}$; $t_{-1}t_0$; t_0t_1; t_1t_2; $t_{-2}t_{-1}t_0$; $t_{-1}t_0t_1$; $t_0t_1t_2$

Table 2: Feature templates for phrase chunking

3 Experimental Results

We evaluate the performance of using several sequence learning models for the Vietnamese NP chunking problem. The data of more than 9,000 sentences is evaluated using an empirical experiment with 5 fold cross validation test. It means we used 1,800 and 7,200 sentences for testing and training the discriminative sequence learning models, respectively. Note that the evaluation method is used the same as CONLL2000 did. We used Precision, Recall, and F-Measure in which Precision measures how many chunks found by the algorithm are correct and the recall is percentage of chunks defined in the corpus that were found by the chunking program.

$$Precision = \frac{\#\text{correct}-\text{chunk}}{\#\text{numberofchunks}}$$
$$Recall = \frac{\#\text{correct}-\text{chunks}}{\#\text{numerofchunksinthecorpus}}$$

$$\text{F} - \text{measure} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

To compute the scores in our experiments, we utilized the evaluation tool (conlleval.pl) which is available in CONLL 2000 (Sang and Buchholz, 2000,).

Figure 2 shows the precision scores of three methods using 5 Folds cross validation test. It reports that the CRF-LBFGS attain the highest score. The SVMs and CRF-SGD are comparable to CRF-LBFGS. The Online Learning achieved the lowest score.

Figure 3 shows the recall scores of three CRFs-LBFGS, CRFs-SGD, SVM, and Online Learning. The results show that CRFs-SGD achieved the highest score while the Online Learning obtained the lowest score in comparison with others.

Figure 4 and Figure 5 show the F-measure and accuracy scores using 5 Folds Cross-validation Test. Similar to these results of Precision and Recall, CRFs-LBFGS was superior to the other ones while the Online Learning method obtained the lowest result.

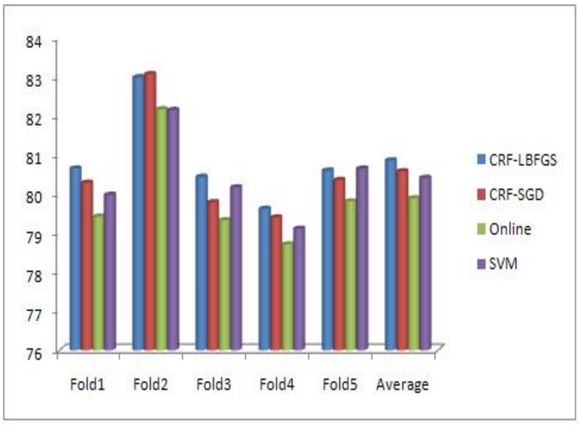


Figure 2: Precision results in 5 Fold cross validation test

Table 3 shows the comparison of three discriminative learning methods for Vietnamese Noun Phrase chunking. We compared the three sequence learning methods including: CRFs using the LBFGS method, CRFs with SGD, and Online Learning. Experiment results show that the CRFs-LBFGS is the best in comparison with others. However, the computational times when training the data is slower than either SGD or Online Learning. The SGD is faster than CRF-LBFS approximately 6 times. The SVM model obtained a comparable results with CRFs models and it was superior to Online Learning. It yields results that were 0.712% than Online Learning. However, the SVM's training process take slower than CRFs and Online Learning. According to our empirical investigation, it takes approximately slower than CRF-SGF, CRF-LBFGS as well as Online Learning.

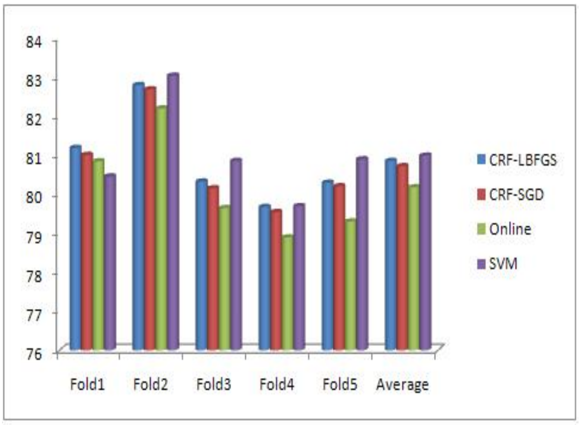


Figure 3: Recall result in 5 Fold cross validation test

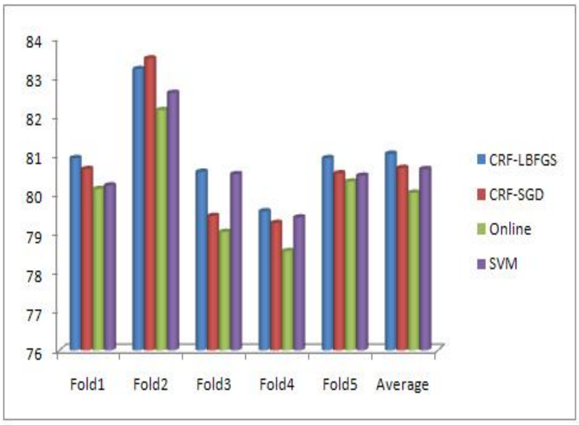


Figure 4: The F-measure results of 5 Folds Cross-validation Test

Note that we used FlexCRFs (Phan, Nguyen, Tu , 2005) for Conditional Random Fields using LBFGS, and for Stochastic Gradient Descent (SGD) we used SGD1.3 which is developed by Leon Bottou [2].

Methods	Precision	Recall	F_1
CRF-LBGS	80.85	81.034	80.86
CRF-SGD	80.74	80.66	80.58
Online-PA	80.034	80.13	79.89
SVM	80.412	80.982	80.638

Table 3: Vietnamese Noun Phrase chunking performance using Discriminative Sequence Learning (CRFs, SVM, Online-PA)

In order to investigate which features are major effect on the discriminative learning models for Vietnamese Chunking problems, we conduct three experiments as follows.

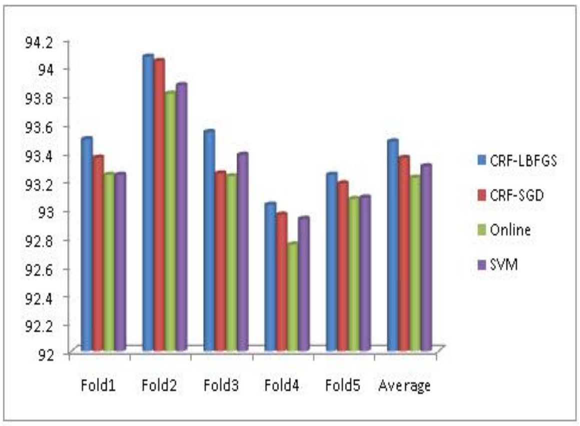


Figure 5: The accuracy scores of four methods with 5 Folds Cross-validation Test

- Cross validation test for three modes without considering the edge features
- Cross validation test for three models without using POS features
- Cross validation test for three models without using lexical features
- Cross validation test for three models without using ”edge features template” features

Note that the computational time of training SVMs model is slow, so we skip considering feature selection for SVMs. We only consider feature selection for CRFs and Online Learning.

Feature Set	LBFGS	SGD	Online
Full-Features	80.86	80.58	79.89
Without-Edge	80.91	78.66	80.13
Without-Pos	62.264	62.626	59.572
Without-Lex	77.204	77.712	75.576

Table 4: Vietnamese Noun Phrase chunking performance using Discriminative Sequence Learning (CRFs, Online-PA)

Table 4 shows that the Edge features have an impact to the CRF-SGD model while it do not affect to the performance of CRFs-LBFGS and Online-PA learning. Table 4 also indicates that the POS features are severed as important features regarding to the performance of all discriminative sequence learning models. As we can see, if one do not use POS features the F1-score of each model is decreased more than 20%. We also remark that the lexical features contribute an important role to the performance of Vietnamese text

[2] http://leon.bottou.org/projects/sgd

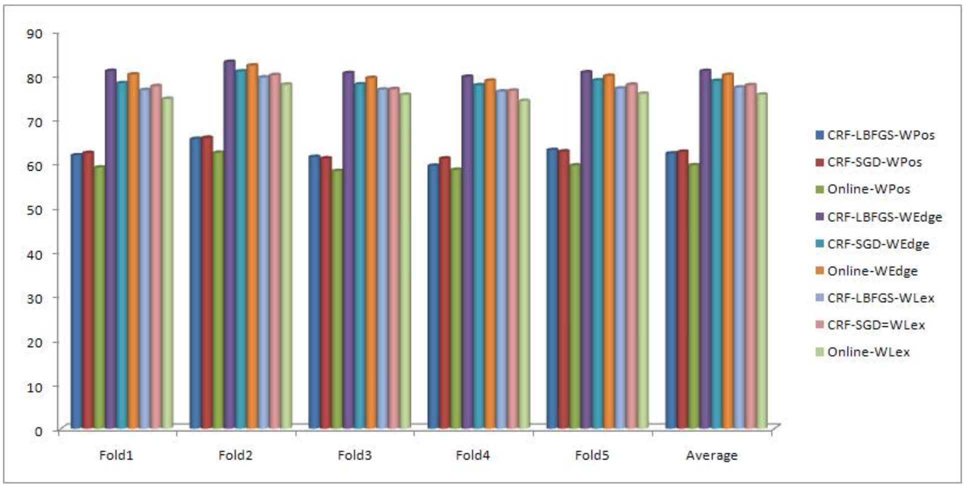


Figure 6: F-measures of three methods with different feature set

chunking. If we do not use lexical features the F1-score of each model is decreased till approximately 3%. In conclusion, the POS features significantly effect on the performance of the discriminative sequence models. This is similar to the note of (Chen, Zhang, and Ishihara, 2006).

Figure 6 reports the F-Measures of using different feature set for each discriminative models. Note that WPos, WLex, and WEdge mean without using Pos features, without using lexical features, and without using edge features, respectively. As we can see, the CRF-LBFGs always achieved the best scores in comparison with the other ones and the Online Learning achieved the lowest scores.

4 Conclusions

In this paper, we report an investigation of developing a Vietnamese Chunking tool. We have constructed an annotation corpus of more than 9,000 sentences and exploiting discriminative learning models for the NP chunking task. Experimental results using 5 Folds cross-validation test have showed that the discriminative models are well suitable for Vietnamese phrase chunking. Conditional random fields show a better performance in comparison with other methods. The part of speech features are known as the most influence features regarding to the performances of discriminative models on Vietnamese phrases chunking.

What our contribution is expected to be useful for the development of Vietnamese Natural Language Processing. Our results and corpus can be severed as a very good baseline for Natural Language Processing community to develop the Vietnamese chunking task.

There are still room for improving the performance of Vietnamese chunking models. For example, more attention on features selection is necessary. We would like to solve this in future work.

Acknowledgments

The constructive comments and helpful suggestions from three anonymous reviewers are greatly appreciated. This paper is supported by JAIST Grant for Research Associates and a part from a national project named Building Basic Resources and Tools for Vietnamese Language and Speech Processing, KC01.01/06-10.

References

M. Collins. 2002. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In Proceedings of EMNLP 2002.

K. Crammer et al. 2006. Online Passive-Aggressive Algorithm. Journal of Machine Learning Research, 2006

W. Chen, Y. Zhang, and H. Ishihara 2006. An empirical study of Chinese chunking. In Proceedings COLING/ACL 2006

Dinh Dien, Vu Thuy 2006. A maximum entropy approach for vietnamese word segmentation. In Proceedings of the IEEE - International Conference on Computing and Telecommunication Technologies RIVF 2006: 248-253

J. Lafferty, A. McCallum, and F. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In the proceedings of International Conference on Machine Learning (ICML), pp.282-289, 2001

N.C. Mai, D.N. Vu, T.P. Hoang. 1997. Foundations of linguistics and Vietnamese. Education Publisher (1997) 142. 152

Thi Minh Huyen Nguyen, Laurent Romary, Mathias Rossignol, Xuan Luong Vu. 2006. A lexicon for Vietnamese language processing. Language Reseourse Evaluation (2006) 40:291-309.

Minh Nghiem, Dien Dinh, Mai Nguyen. 2008. Improving Vietnamese POS tagging by integrating a rich feature set and Support Vector Machines. In Proceedings of the IEEE - International Conference on Computing and Telecommunication Technologies RIVF 2008: 128–133.

X.H. Phan, M.L. Nguyen, C.T. Nguyen. FlexCRFs: Flexible Conditional Random Field Toolkit. http://flexcrfs.sourceforge.net, 2005

T. Kudo and Y. Matsumoto. 2001. Chunking with Support Vector Machines. The Second Meeting of the North American Chapter of the Association for Computational Linguistics (2001)

F. Sha and F. Pereira. 2005. Shallow Parsing with Conditional Random Fields. Proceedings of HLT-NAACL 2003 213-220 (2003)

C.T. Nguyen, T.K. Nguyen, X.H. Phan, L.M. Vietnamese Word Segmentation with CRFs and SVMs: An Investigation. 2006. The 20th Pacific Asia Conference on Language, Information, and Computation (PACLIC), 1-3 November, 2006, Wuhan, China

Tjong Kim Sang and Sabine Buchholz. 2000. Introduction to the CoNLL-2000 Shared Task: Chunking. Proceedings of CoNLL-2000 , Lisbon, Portugal, 2000.

V. Vapnik. 1995. The Natural of Statistical Learning Theory. New York: Springer-Verlag, 1995.