A Semi-Supervised Learning Method for Vietnamese Part-of-Speech Tagging

Le Minh Nguyen Bach Ngo Xuan Cuong Nguyen Viet Minh Pham Quang Nhat
Akira Shimazu
School of Information Science, JAIST
1-1 Asahidai, Nomi, Ishikawa 923-1292, Japan
{nguyenml,bachnx,cuongnv,minhpqn,shimazu}@jaist.ac.jp

Abstract

This paper presents a semi-supervised learning method for Vietnamese part of speech tagging. We take into account two powerful tagging models including Conditional Random Fields (CRFs) and the Guided Online-Learning models (GLs) as base learning models. We then propose a semi-supervised learning tagging model for both CRFs and GLs methods. The main idea is to use of a word-cluster model as an associate source for enrich the feature space of discriminate learning models for both training and decoding processes. Experimental results on Vietnamese Tree-bank data (VTB) showed that the proposed method is effective. Our best model achieved accuracy of 94.10% when tested on VTB, and 92.60% an independent test.

1 Introduction

Part of speech tagging is the fundamental task of natural language processing. Most of works on Vietnamese part of speech tagging is based on statistical machine learning models in which a large of annotated corpora is required to train the tagging models. Previous Vietnamese tagging systems mainly based on maximum entropy models [10, 14, 6], which consider the Vietnamese part of speech tagging problem as sequence learning. The previous systems were trained and estimated on the VTB data [11].

Discriminate learning models are well suitable for the part of speech tagging task [2, 5, 3]. However, the process of annotating data is expensive, and requires much effort. To deal with this limitation, we select the semi-supervised learning approach, which relies on the use of unsupervised data to improve the performance of part of speech tagging. Our motivation is generally based on the observation that the tagger fail when tagging sentences from domains other than the training data. One solution is to use a large un-annotated corpus to enrich feature-set in the discriminative models, in order to establish correlations with features of test data. A simple method is to use the brown-clustering method [1, 9, 7, 4] to cluster words and map them as features for improving the performance of discriminative learning models.

The original idea of combining word clusters with discriminative learning has been previously explored by [9], which is mainly applied for Named Entity recognition. For more complex problems, Koo and Collins [4] have incorporated word-cluster models with discriminative learning for dependency tree parsing. They showed that word-cluster models are very suitable for this task and their system significantly outperformed the state of the art results. Nguyen et al.[12] applied word-cluster models to automatically generating table-of-contents. They showed that word-cluster models are useful for improving the accuracy of generating table of content.

Although the word-cluster models are effective for English processing applications, their advantage for Vietnamese processing, especially for part of speech tagging still remains in guesstion. In this paper, we first introduce a semi-supervised learning framework for Vietnamese part of speech tagging. We investigate the use of a word-cluster model constructed from large text documents, with CRFs based tagging and perceptron-style tagging system. We will show that the proposed semi-supervised learning model is effective for improving the performance of Vietnamese part of speech tagging. Our contribution in this paper is application of the two learning models to Vietnamese part of speech tagging. According to our understanding so far, there are no works reporting the performance of incorporating word-cluster models with CRFs and GLs in Vietnamese part of speech tagging problems.

The paper consists of three parts. The first section introduces the background of part of speech tagging for Vietnamese. The second section presents our tagging models. The third section shows experimental results and provide a discussion. The final section gives the conclusion and future work.

978-0-7695-4213-3/10 $25.00 © 2010 IEEE
DOI 10.1109/KSE.2010.35

2 Background

In this section, we briefly introduce Conditional Random Fields [5] for part of speech tagging. Second, we present a guided online learning model which is an instantance of Voted-Perceptron model [3] for tagging that utilizes the advantage of the bidirectional inference methods. The advantage of the online learning model is that it can effectively learn both the direction and models for tagging. The technique is mainly based on the guided learning approach [13].

2.1 Characteristics 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 five parts: first consonant, secondary vowel, main vowel, last consonant and a tone mark. For instance, the syllable "tuan" (week) has a tone mark (grave accent), a first consonant (t), a secondary vowel (u), a main vowel (a) and a last consonant (n). However, except for main vowel which 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 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 ways of constructing words from syllables, we can classify them into three categories: single words, complex words and reduplicative words [8].

The pasts of speeches (POS) of each word in Vietnamese are mainly sketched as in Table 1. The definition of these tags are based on Vietnamese treebank [11].

2.2 Conditional Random Fields for Tagging

Conditional Random Fields (CRFs) [5] are undirected graphical models used to calculate the conditional 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 a finite state machine (FSMs). Training CRFs is commonly performed by maximizing the likelihood function with respect to the training data using advanced convex optimization techniques like L-BFGS. 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 Guided Learning for Bi-directional Tagging

There are two decoding methods for part of speech tagging, which consist of the single directional method (such as the left to right and the right to left direction method) and the bi-directional method. The bi-directional methods have shown that they are able to effectively improve the performance of tagging [13] because this method allows learning integration of individual classification and order selection simultaneously. This has been reported to provide the best result when tested on Penn treebank data [13].

We sketch the bidirectional inference algorithm as follows.

2.3.1 Bidirectional Inference Algorithm

Assume that we are given a sequence of tokens $t_1, t_2, ...t_N$, for each token we have to assign $l_i \in L$, with L being the label set. We call a subsequence $t_i, ..., t_j$ a span [i,j]. Each span s[i,j] is associated with one or more hypotheses, which have length (j-i) over L. The labels located at the boundaries of a hypothesis sequence are used as context for labeling tokens outside the span s. Using trigram model, to predict the label L_i, we can use the two labels L_{i+1}, L_{i+2} of the $s[i+1, j]$ if they have already been tagged. They are similar to the two labels L_{i-2}, and L_{i-1}. We will refer to the left two labels as the left interface I_{left}, and to the right two labels as the right interface I_{right}. Let the boundaries of a span s with $b = (I_{left}; I_{right})$, b contain the labels relevant for the tagging of neighboring tokens. We use a matrix $M_p(s)$ $s = (I_{left}, I_{right})$ is the set of all hypotheses associated with p that are compatible with I_{left} and I_{right}.

For a span p and a state s, we denote the associated top hypothesis as

$$s.T = \arg\max_{h \in M_p(s)} V(h),$$

where V is the score of a hypothesis. The top state for p can be defined as below.

$$p.S = \arg\max_{s: M_p(s) \neq \theta} V(h)$$

Spans are started and grown by means of tagging actions. Three kinds of actions are available: it is possible to start a new span by labeling a token with no context, or expand an existing span by labeling an adjacent token, or merge two spans by labeling the token between them. In this last case, the two originating spans would be subsequences of the resulting span, and the labeling action of the token between the spans will use both right and left context information. For each hypothesis h associated with a span s, we maintain its most recent tagging action $h.A$, and the hypotheses, if any, that have been used as left context $h.S_L$ and right $h.S_R$

No.	Category	Description	No.	Category	Description
1.	Np	Proper noun	10.	M	Numeral
2.	Nc	Classifier	11.	E	Preposition
3.	Nu	Unit noun	12.	C	Subordinating conjunction
4.	N	Common noun	13.	CC	Coordinating conjunction
5.	V	Verb	14.	I	Interjection
6.	A	Adjective	15.	T	Auxiliary, modal words
7.	P	Pronoun	16.	Y	Abbreviation
8.	R	Adverb	17.	Z	Bound morpheme
9.	L	Determiner	18.	X	Unknown

Table 1. Part-of-Speech in Vietnamese

We can now define the score function for hypotheses in a recursive fashion:

$$V(h) = V(h^*_L(h)) + V(h^*_R(h)) + U(h.A)$$

In which

$$U(h.A) = w \cdot f(h.A)$$

Algorithm 1 shows the prototype of the bi-directional decoder algorithm for part of speech tagging. For details please refer to [13].

Input: $S = (w_i), i = 1, 2, \ldots, n$
Beam width B
Weight vector w

1 Initialize P—the set of accepted spans
2 Initialize Q—the queue of candidate spans
3 **for** $t \leftarrow 1, 2, \ldots$ **do**
4 Span $p' \leftarrow \arg\max_{p \in Q} U(p.s.T.A)$
5 Update P with p'
6 Update Q with p' and P
7 **end**

Algorithm 1: Bidirectional decoder algorithm

2.3.2 Learning Algorithm

In this section, we would like to summarize the guided learning [13], a perceptron-like algorithm to learn the weight vector w as shown in Algorithm 2. Assume that $p'.G$ represents the gold standard hypothesis on a span p'. For each input sequence X_r and a gold standard sequence of tagging Y_r, P and Q are initialized, which is the same as in Algorithm 1. Line 8 shows how we select the span for the next moves. If $p'.S.T$, the top hypothesis of the selected span p', is compatible with the gold standard, then P and Q are updated (as shown in Line 9 and Line 10). Otherwise, we update the weight vector in the Perceptron style, by promoting the features of the gold standard action, and demoting the feature of the action of the top hypothesis (Line 13 and Line 14). All elements in the queue Q are generated with P and the updated vector w. This process starts by removing all the elements in Q, and then generate hypotheses for all the possible spans based on the context span in P. Hypothesis scores and action scores are calculated based on w. Note that in Algorithm 2 two scores are maintained: the score of the action represents the confidence for the next move, and the score of the hypothesis represents the overall quality of a partial result. The selection for the next move depends on only the score of action. The score of a hypothesis is used to maintain top partial results for each span. See the paper [13] for the details of the algorithm and its soundness.

Data: $S = (X_r, Y_r), r = 1, 2, \ldots, R$
Beam width B

1 $w \leftarrow 0$
2 **for** $i \leftarrow 1$ **to** I **do**
3 **for** $r \leftarrow 1$ **to** R **do**
4 Initialize P—the set of accepted spans
5 Initialize Q—the queue of candidate spans
6 **for** $t \leftarrow 1, 2, \ldots$ **do**
7 Span $p' \leftarrow \arg\max_{p \in Q} U(p.s.T.A)$
8 **if** $p'.S.T = p'.G$ **then**
9 Update P with p'
10 Update Q with p' and P
11 **end**
12 **else**
13 Promote($w, f(p'.G.A)$)
14 Demote($w, f(p'.S.T.A)$
15 Re-generate Q with w and P
16 **end**
17 **end**
18 **end**
19 **end**

Algorithm 2: Guided learning algorithm

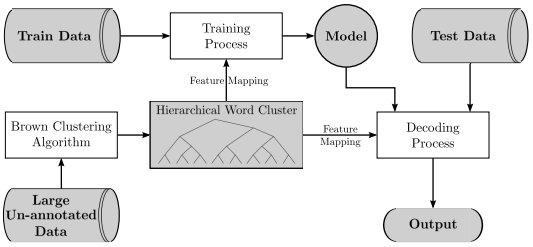

Figure 1. A semi-supervised learning framework for part of speech tagging

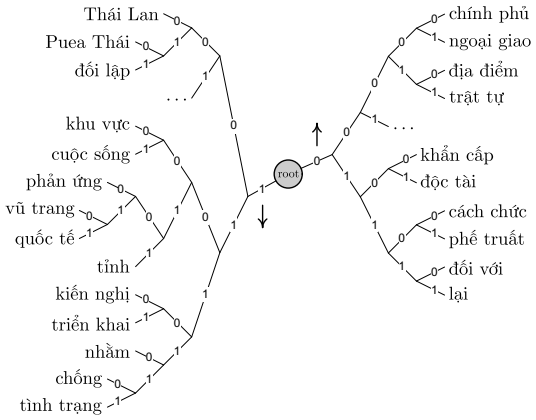

Figure 2. An example of a hierarchical clustering. Each word at a leaf is encoded by a bit string with respect to the path from the root, where 0 indicates an "up" branch and 1 indicates a "down" branch.

3 A Semi-Supervised Discriminative Learning Models for Tagging

In this section, we present a semi-supervised learning framework for part of speech tagging. We present a general framework for incorporating word-cluster modes to discriminative learning. Figure 1 shows our semi-supervised learning framework for discriminative learning models.

Larger un-annotated text documents are clustered using the brown-clustering method to obtain word-cluster models. A word-cluster model is then used to enrich feature space for discriminate learning models in both the training and testing process.

3.1 The Brown Algorithm

The Brown algorithm is a hierarchical agglomerative word clustering algorithm [1]. The input of this algorithm is a large sequence of words $w_1, w_2, \ldots, w_n$, which are extracted from raw texts. The output of this algorithm is a hierarchical clustering of words—a binary tree—wherein a leaf represents a word, and an internal node represents a cluster containing the words in the sub-tree, whose root is that internal node.

This algorithm uses contextual information—the next word information—to represent properties of a word. More formally, $C(w)$ denotes the vector of properties of w (or w's context). We can think of our vector for w_i as counts, for each word w_j, of how often w_j followed w_i in the corpus:

$$C(w_i) = (|w_1|, |w_2|, \ldots, |w_n|)$$

$C(w_i)$ is normalized by the count of w_i, and then we would have a vector of conditional properties $P(w_j|w_i)$. The clustering algorithm used here is HAC-based. Therefore, at each iteration, it must determine which two clusters are combined into one cluster. The metric used for that purpose is the minimal loss of average mutual information.

Figure 2 shows a portion of a hierarchical clustering, which is derived from a small portion of text, which contains 12 sentences and 182 distinct words. This portion of text is about the political situation in Thailand.

From this tree, we can freely get a cluster of words by collecting all words at the leaves of the sub-tree, whose root is a chosen internal node. For instance, some clusters are shown in 2.

To use word cluster information in our model at several levels of abstraction, we encode each word cluster by a bit string that describes the path from the root to the chosen internal node. The path is encoded as follows: we start from the root of the hierarchical clustering, "0" is appended to the binary string if we go up, and "1" is appended if we go down. For instance, to encode the above three clusters, we use the following bit strings "100", "110", "010", and "1111", respectively. If we want to use a higher level of abstraction, we can simply combine the clusters that have the same prefix. For instance, if we need only two clusters, we can use the prefix with the length of 1. In that situation, all the words in the left sub-tree are in a cluster encoded by "1", and all the words in the right sub-tree are in another cluster encoded by "0".

3.2 Feature Set

3.2.1 Basic Feature Set

the feature set is designed through feature templates, which are 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., "$l = V$" and "$l' = N$"). 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. Table 2 describes both edge and observation feature templates. Statistics for observation features are identities of words surrounding the current position, such as words 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).

3.2.2 Feature Set Using Word-Cluster

In addition to the baseline features presented in the previous section, we used word-cluster to enrich the feature space. Each word in the training data and testing data is mapped to a bit string. Each cluster of words is represented by a bit string as described in Section 3.1. For example, the word "cach_chuc" and "phe_truat" are represented as "01100" and "01101", respectively. The cluster information of a word is also represented by the bit string of the cluster containing that word. We create an indicator function for each cluster, and use it as a selection feature:

$$f_{110101}(w) = \begin{cases} 1 & \text{if } w \text{ has bit string } 110101, \\ 0 & \text{otherwise.} \end{cases}$$

We also used the conjunction between bit representation with previous tags (window size 2) for our learning models.

4 Experimental Results

In order to build word-cluster models for Vietnamese language processing, we crawled raw texts from VietnamNet online newspaper[1], which include 65,112 articles and 2,226,169 sentences. We ran the Vietnamese word segmentation (published on Vietnam National Project[2]) to obtain approximately 4,4 millions words (except non-words token). After that, we applied the brown-clustering method to obtain hierarchical representation for each word. To illustrate the performance of our tagging systems, we used the training data on the VTB corpus [11] which includes approximately 20,000 annotated sentences collected from the Youth online daily newspaper. The minimal and maximal sentence lengths are 2 words and 105 words respectively.

We randomly take 80% of the corpus for training and 20% for testing. We compare the Conditional Random Fields, and Guided Learning and their combination with word-cluster set. In addition, we also test the proposed systems on the test data, which is a very different domain in comparison with the training data. The number of test sentences is 100 sentences. Table 3 shows the results of CRFs, Guided Learning, CRFs and Guided Online Learning semi-supervised learning model with word-cluster features.

Methods	In domain	Out domain
CRFs	91.28	89.74
Semi-CRFs	91.45	90.27
Guided Learning	93.20	91.50
Semi-Guided Learning	94.10	92.20

Table 3. Vietnamese part of speech tagging performance using Discriminative Sequence Learning and its combination with word-cluster

Table 3 demonstrates that semi-supervised learning models using word-clusters improve the performance of both CRFs and Online learning models for the Vietnamese tagging problem. When we tested the proposed models on a test domain which different from the domain of training data, the errors in tagging using word-cluster are much reduced. As we can see in Table 3, Semi-CRFs lead to improvement of 0.53% in comparison with CRFs. Meanwhile, Semi-CRFs improve only 0.17% on evaluating the test on In Domain. In addition, Semi-GLs also work effectively with the test in Out Domain. It improves 0.7% and 0.7% in comparison with GLs when evaluating on "Out Domain" and "In Domain", respectively.

This was because some unknown words can be predicted correctly due to their bit string appeared in the training data. This is the advantage of using word-cluster models for tagging problem.

The results also indicate that online learning with bi-directional decoding can significantly outperform CRFs. Figure 3 demonstrates that GLs won CRFs on both test sets. This can be explained that GLs utilizing the labels predicted in both directions to reduce ambiguities. It clearly demonstrates that GLs and Semi-GLs are effective for tagging those sentences in the same training domain and different domains.

On of the advantages of GLs over CRFs is that it is very easy to implement, and it is faster that other learning methods.

5 Conclusion

In this paper, we originally introduce the guided learning models for Vietnamese part of speech tagging and show that this model significantly outperformed the Conditional

[1] http://vietnamnet.net

[2] http://vlsp.vietlp.org:8080/demo/

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;
l	Is w_0 punctuation? Is w_0 capitalized?

Table 2. the feature set for part of speech tagging

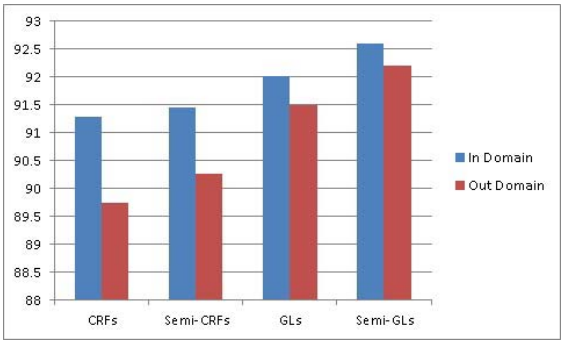

Figure 3. A comparison of discriminative learning models (CRFs and GLs) on two test domains

Random Fields on the same data and feature set. In addition, we propose a simple unsupervised learning model for Vietnamese part of speech tagging, which used word-cluster models are a feature mapping process, to enrich the space of features in both CRFs and GLs models. Experimental results on the VTB data showed that word-cluster models can be significantly improved the accuracy of discriminative learning models. In future work, we will investigate how word-cluster models can be applied for other Vietnamese language processing applications such as shallow parsing and Named Entity recognition.

Acknowledgments

This paper is supported by Grants-in-Aid for Scientific Research (22700139).

References

[1] P. F. Brown, P. V. Desouza, R. L. Mercer, and J. C. Lai. Class-based n-gram models of natural language. *Computational Linguistics*, 18(4):467–479, 1992.

[2] M. Collins. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In *Proceedings of the ACL-EMNLP 2002*, pages 1–8, Morristown, NJ, USA, 2002.

[3] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer. Online passive-aggressive algorithms. *Journal of Machine Learning Research*, 7:551–585, 2006.

[4] T. Koo, X. Carreras, and M. Collins. Simple semi-supervised dependency parsing. In *Proceedings of ACL-HLT 2008*, pages 595–603, Columbus, Ohio, USA, 2008.

[5] J. D. Lafferty, A. McCallum, and F. C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of ICML 2001*, pages 282–289, San Francisco, CA, USA, 2001.

[6] H. P. Le, A. Roussanaly, T. M. H. Nguyen, and M. Rossignol. An empirical study of maximum entropy approach for part-of-speech tagging of vietnamese texts. In *Proceedings of TALN 2010*, Montreal, Canada, 2010.

[7] P. Liang and M. Collins. Semi-supervised learning for natural language. Master's thesis, MIT, 2005.

[8] N. C. Mai, D. N. Vu, and T. P. Hoang. *Foundations of Linguistics and Vietnamese*. Education Publisher, 1997.

[9] S. Miller, J. Guinness, and A. Zamanian. Name tagging with word clusters and discriminative training. In *Proceedings of HLT-NAACL 2004*, pages 337–342, Boston, Massachusetts, USA, 2004.

[10] M. Nghiem, D. Dinh, and M. Nguyen. Improving vietnamese pos tagging by integrating a rich feature set and support vector machines. In *Proceedings of RIVF 2008*, pages 128–133, 2008.

[11] P. T. Nguyen, X. L. Vu, T. M. H. Nguyen, V. H. Nguyen, and H. P. Le. Building a large syntactically-annotated corpus of vietnamese. In *Proceedings of the Third Linguistic Annotation Workshop, ACL-IJCNLP '09*, pages 182–185, Suntec, Singapore, 2009.

[12] V. C. Nguyen, L. M. Nguyen, and A. Shimazu. A semi-supervised approach for generating a table-of-contents. In *Proceedings of RANLP 2009*, pages 313–318, Borovets, Bulgaria, 2009.

[13] L. Shen, G. Satta, and A. Joshi. Guided learning for bidirectional sequence classification. In *Proceedings of ACL 2007*, pages 760–767, Prague, Czech Republic, 2007.

[14] T. O. Tran, A. C. Le, and Q. T. Ha. Improving vietnamese word segmentation and pos tagging using mem with various kinds of resources. *Journal of Natural Language Processing in Japan*, 2010.