VNU Journal of Science: Comp. Science & Com. Eng., Vol. 32, No. 3 (2016) 11–25

An Experimental Investigation of Part-Of-Speech Taggers for Vietnamese

Nguyen Tuan Phong[1], Truong Quoc Tuan[1], Nguyen Xuan Nam[1], Le Anh Cuong[2,*]

[1]*Faculty of Information Technology, VNU University of Engineering and Technology, No. 144 Xuan Thuy Street, Dich Vong Ward, Cau Giay District, Hanoi, Vietnam*

[2]*Faculty of Information Technology, Ton Duc Thang University, No. 19 Nguyen Huu Tho Street, Tan Phong Ward, District 7, Ho Chi Minh City, Vietnam*

Abstract

Part-of-speech (POS) tagging plays an important role in Natural Language Processing (NLP). Its applications can be found in many other NLP tasks such as named entity recognition, syntactic parsing, dependency parsing and text chunking. In the investigation conducted in this paper, we utilize the techniques of two widely-used toolkits, ClearNLP and Stanford POS Tagger, and develop two new POS taggers for Vietnamese, then compare them to three well-known Vietnamese taggers, namely JVnTagger, vnTagger and RDRPOSTagger. We make a systematic comparison to find out the tagger having the best performance. We also design a new feature set to measure the performance of the statistical taggers. Our new taggers built from Stanford Tagger and ClearNLP with the new feature set can outperform all other current Vietnamese taggers in term of tagging accuracy. Moreover, we also analyze the affection of some features to the performance of statistical taggers. Lastly, the experimental results also reveal that the transformation-based tagger, RDRPOSTagger, can run faster than any statistical tagger significantly.

Received March 2016, Revised May 2016, Accepted May 2016

Keywords: Part-of-speech tagger, Vietnamese.

1. Introduction

In Natural Language Processing, part-of-speech tagging is the process to assign a part-of-speech to each word in a text according to its definition and context. POS tagging is a core task of NLP. The part-of-speech information can be used in many other NLP tasks, including named entity recognition, syntactic parsing, dependency parsing and text chunking. In common languages such as English and French, studies in POS tagging are very successful. Recent studies for these languages [1-5] can yield state-of-the-art results at approximately 97-98% for overall accuracy. However, for less common languages such as Vietnamese, current results are not as good as for Western languages. Recent studies on Vietnamese POS tagging such as [1, 2] can only achieves approximately 92-93% for precision.

Several POS tagging approaches have been studied. The most common ones are

[*] Corresponding author. Email.: leanhcuong@tdt.edu.vn

stochastic tagging, rule-based tagging and transformation-based tagging whereas the last one is a combination of the others. All of these three approaches treat POS tagging as a supervised problem that requires a pre-annotated corpus as training data set. For English and other Western languages, almost studies that provide state-of-the-art results are based on the supervised learning. Similarly, the most widely-used taggers for Vietnamese, JVnTagger [3], vnTagger [1] and RDRPOSTagger [2], also treat POS tagging as a supervised learning problem. While JVnTagger and vnTagger are stochastic-based tagger, RDRPOSTagger implements a transformation-based approach. Although these three taggers are reported to have the highest accuracies for Vietnamese POS tagging, they can only give the precision of 92-93%. Meanwhile, two well-known open-source toolkits, ClearNLP [4] and Stanford POS Tagger [5], which use stochastic tagging algorithms can provide overall accuracies of over 97% for English. It would be unfair to compare the results for two different languages because they have distinct characteristics. Therefore, our questions are *"How well can the two international toolkits perform POS tagging for Vietnamese?"* and *"Which is the most effective approach for Vietnamese part-of-speech tagging?"*. The purpose of the investigation conducted in this paper is to answer those questions by doing a systematic comparison of the taggers. Beside the precision of taggers, their tagging speed is also considered because many recent NLP tasks have to deal with very large-scale data in which speed plays a vital role.

For our experiments, we use Vietnamese Treebank corpora [6] which is the most common corpus and has been utilized by many studies on Vietnamese POS tagging and is one resource from a national project named "Building Basic Resources and Tools for Vietnamese Language and Speech Processing" (VLSP)[1]. Vietnamese Treebank contains about 27k POS-tagged sentences. In spite of its popularity, there have been several errors in this data that can draw the precision of taggers. All of those errors that we detected are also reported in this paper.

By using 10-fold cross-validation method on the configured corpus, it is revealed that the new taggers we built from ClearNLP and Stanford POS Tagger produce the most accurate results at 94.19% and 94.53% for precision, which also are the best Vietnamese POS tagging results known to us. Meanwhile, the highest tagging speed belongs to the transformation-based tagger, RDRPOSTagger, which can assign tags for over 161k words per second in average while running on a personal computer.

The remainder of this paper is organized as follows. In section 2, we briefly introduce general knowledge about the main approaches that have been applied in POS tagging task. We also give some information about particular characteristics of Vietnamese language and the experimental data, Vietnamese Treebank corpora. Section 3 represents the methods used by the POS taggers.

In section 4, we talk about the main contribution of this paper including the error fixing process for the experimental data, the experimental results on the taggers and the comparison of their accuracies and tagging speeds. Finally, we conclude this paper in section 5.

[1] http://vlsp.vietlp.org:8080/demo/?page=home

2. Background

This section provides some background information of part-of-speech tagging approaches that have been used so far. The related works are also covered. Moreover, we also give some details about Vietnamese language and Vietnamese Treebank.

2.1. Approaches for POS tagging

Part-of-speech tagging is commonly treated as a supervised learning problem. Each POS tagger takes the information from its training data to determine the tag for each word in input text. In most cases, a word might have only one possible tag.

The other case is that a word has several possible tags; or a word has not appeared in the lexicon extracted from the training data. The process to choose the right tag for a word in these cases is based on which kind of used tagging algorithm. There are three main kinds of tagging approaches within POS tagging, which are stochastic tagging, rule-based tagging and transformation-based tagging.

Stochastic (probabilistic) tagging approach is one of the most widely-used ones in recent studies for POS tagging. The general idea of stochastic taggers is that they make use of training corpus to determine the probability of a specific word having a specific tag in a given context. Common methods of stochastic approach are Maximum Entropy (MaxEnt), Conditional Random Fields (CRFs), Hidden Markov Models (HMMs). Many studies on English POS tagging using stochastic approaches can gain state-of-the-art results, such as [5, 4, 7].

Rule-based tagging is actually different from stochastic tagging. Rule-based tagging algorithm uses a set of hand-written rules to determine the tag for each word. This leads to a fact that this set of rules must be properly written and checked by experts on linguistic.

Meanwhile, transformation-based tagging is a combination of the features of the two algorithms above. This algorithm applies disambiguation rules like the rule-based tagging, but these rules are not hand-written. They are automatically extracted from the training corpus. Taggers using this kind of algorithm are usually referred to Brill's one [8]. There are three main steps in his algorithm. Firstly, the tagger initially assigns for each word in the input text with the tag which is the most frequent for this word in the lexicon extracted from the training corpus. After that, it traverses through a list of transformation rules to choose the rule that enhances tagging accuracy the most. Then this transformation rule will be applied to every word. The loop through three stages is continued until it optimizes the tagging accuracy.

For all of those approaches listed above, a pre-annotated corpus is prerequisite. On the other hand, there is also unsupervised POS tagging algorithm [9, 10] that does not require any pre-tagged corpus.

For Vietnamese POS tagging, Tran [11] compares three tagging methods which are CRFs-based, MEMs-based and SVM-based tagging. However, the comparison does not contain terms of unknown words accuracy and tagging speed. Moreover, all of those methods are based on stochastic tagging.

It is necessary to systematically compare all of those characteristics of the taggers in a same evaluation scheme and also the accuracies of different kinds of approach to find out the most accurate one for Vietnamese POS tagging.

2.2. *Vietnamese language*

In this section, we talk about some specific characteristics of Vietnamese language compared to the Western languages and also some information of Vietnamese Treebank, the corpus which we use for experiments.

2.2.1. The language

Vietnamese is an Austroasiatic language and the national and official language of Vietnam. It is the native language of Kinh people. Vietnamese is spoken throughout the world because of Vietnamese emigration. The Vietnamese alphabet in use today is a Latin alphabet with additional diacritics and letters.

In Vietnamese, there is no word delimiter. Spaces are used to separate the syllables rather than the words. For example, in the sentence *"[học sinh] [học] [sinh học]"* (*"students study biology"*), there are two times that *"học sinh"* appears, the first space between *"học sinh"* is the separation of two syllables of the word *"học sinh"* (*"students"*), however, the second one is not.

Vietnamese is an inflectionless language whose word forms never change as in occidental languages. There are many cases in that a word has more than one part-of-speech tags in different contexts. For instance, in the sentence *"[học sinh] [ngồi] [quanh] [bàn]$_1$ [để] [bàn]$_2$ [về] [bài] [toán]"* (*"students sit around the [table]$_1$ in order to [discuss]$_2$ about a Math exercise"*), the first word *bàn* is a noun but the second one is a verb. Part-of-speech for Vietnamese words is usually ambiguous so that they must be classified based on their syntactic functions and meaning in their current context.

2.2.2. Vietnamese Treebank

Vietnamese Treebank [6] is the largest annotated corpora for Vietnamese. It is one of the resources from the KC01/06-10 project named "Building Basic Resources and Tools for Vietnamese Language and Speech Processing" which belongs to the National Key Science and Technology Tasks for the 5-Year Period of 2006-2010. The first version of the treebank consists of 10,165 sentences which are manually segmented and POS-tagged. This number in the current version of the treebank is increased to 27,871 annotated sentences[2]. The raw texts of the treebank are collected from the social and political sections of the Youth online daily newspaper. The minimal and maximal sentence lengths are 1 words and 165 words respectively.

The tagset designed for Vietnamese Treebank is presented in Table 1. Beside these eighteen basic tags, there are also compound tags such as *Ny* (abbreviated noun), *Nb* (foreign noun) or *Vb* (foreign verb).

3. Method analysis

This section provides information about the general methods used by current Vietnamese POS taggers and two taggers for common languages. While RDRPOSTagger uses a transformation-based learning approach, all of four other taggers, ClearNLP, Stanford POS Tagger, vnTagger and JVnTagger, are stochastic-based taggers using either MaxEnt, CRFs models or support vector classification.

[2] http://vlsp.vietlp.org:8080/demo/?page=resources

Table 1. Vietnamese tagset

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

3.1. *Current Vietnamese POS taggers*

3.1.1. JVnTagger

JVnTagger is a stochastic-based POS tagger for Vietnamese and is implemented in Java. This tagger is based on CRFs and MaxEnt models. JVnTagger is a branch product of VLSP project and also a module of JVnTextPro, a widely used toolkit for Vietnamese language processing developed by Nguyen and Phan [3]. This tagger is also called by the other name, VietTagger.

There are two kinds of feature used in JVnTagger, which are context features for both CRFs and MaxEnt models and an edge feature for CRFs model as listed in Table 2. Both models of JVnTagger use 1-gram and 2-gram features for predicting tags of all words. For unknown words, this toolkit uses some rules to detect whether each word is in a specific form or not to determine its part-of-speech tag.

Additionally, there is a particular feature extracted by looking up the current word in a tags-of-word dictionary which contains possible tags of over 31k Vietnamese words extracted before. This feature applies for both the current word, the previous and the next words. Besides, in Vietnamese, repetitive word is a special feature, therefore, JVnTagger adds full-repetitive and partial-repetitive word features to enhance the accuracy of predicting tag *A* (adjective) as well. Word prefix and suffix are also vital features in POS tagging task of many other languages.

The CRFs model of JVnTagger had been trained by FlexCrfs toolkit [12]. Due to the nature of CRFs model, there is an edge feature extracted directly by FlexCrfs as described in Table 2.

The F-measure results of JVnTagger are reported at 90.40% for CRFs model and 91.03% for MaxEnt model using 5-fold cross-validation evaluation on Vietnamese Treebank corpus of over 10k annotated sentences.

3.1.2. vnTagger

vnTagger[3] is also a stochastic-based POS tagger for Vietnamese which is developed by Le [1]. The main method of this tagger is Maximum Entropy. vnTagger is written in

[3] http://mim.hus.vnu.edu.vn/phuonglh/softwares/vnTagger

Table 2. Default feature set used in JVnTagger.
w_i: the word at position i in the 5-word window. t_i: the POS tag of w_i

Model	Type	Template
MaxEnt and CRFs	Lexicon	$w_{\{-2,-1,0,1,2\}}$ (w_{-1}, w_0), (w_0, w_1) f
	Binary	w_i contains all uppercase characters or not $(i = -1, 0)$, w_i has the initial character uppercase or not $(i = -1, 0)$, w_i is a number or not $(i = -1, 0, 1)$, w_i contains numbers or not $(i = -1, 0, 1)$, w_i contains hyphens or not $(i = -1, 0)$, w_i contains commas or not $(i = -1, 0)$, w_i is a punctuation mark or not $(i = -1, 0, 1)$ possible tags of w_i in dictionary $(i = -1, 0, 1)$,
	Vietnamese specialized features	w_0 is full repetitive or not, w_0 is partial repetitive or not, the first syllable of w_0, the last syllable of w_0
CRFs	Edge feature	(t_{-1}, t_0)

Java and its architecture is mainly based on the basis of Stanford POS Tagger [5].

There are two kinds of feature used in the MaxEnt model of this tagger, which are presented in Table 3. The first one is the set of features used for all words. This tagger uses a one-pass, left-to-right tagging algorithm, which only make use of information from history. It only captures 1-gram features for words in a window of size 3, and the information of the tags in the left side of the current words. The other kind of feature is used for predicting tags of unknown words. These features mainly help to catch the word shape.

The highest accuracy is reported at 93.40% in overall and 80.69% for unknown words when using 10-fold cross-validation on Vietnamese Treebank corpus of 10,165 annotated sentences.

3.1.3. RDRPOSTagger

RDRPOSTagger [2] is a Ripple Down Rules-based Part-Of-Speech Tagger which is based upon transformation-based learning, a method which is firstly introduced by Eric Brill [8] as mentioned above. It is developed by Nguyen and hosted in Sourceforge[4]. For English, it reaches accuracy figures up to 96.57% when training and testing on selected sections of the Penn WSJ Treebank corpus [13]. For Vietnamese,

[4] http://rdrpostagger.sourceforge.net

Table 3. Default feature set used in vnTagger

Usage	Template
All words	$w_{\{-1,0,1\}}$
	$t_{-1}, (t_{-2}, t_{-1})$
Unknown words	w_0 contains a number or not,
	w_0 contains an uppercase character or not,
	w_0 contains all uppercase characters or not,
	w_0 contains a hyphen or not,
	the first syllable of w_0,
	the last syllable of w_0,
	conjunction of the two first syllables of w_0,
	conjunction of the two last syllables of w_0,
	number of syllables in w_0

it approaches 93.42% for overall accuracy using 5-fold cross-validation on Vietnamese Treebank corpus of 28k annotated sentences. This toolkit has both Java-implemented and Python-implemented versions.

The difference between the approach of RDRPOSTagger to Brill's is that RDRPOSTagger exploits a failure-driven approach to automatically restructure transformation rules in the form of a Single Classification Ripple Down Rules (SCRDR) tree. It accepts interactions between rules, but a rule only changes the outputs of some previous rules in a controlled context. All rules are structured in a SCRDR tree which allows a new exception rule to be added when the tree returns an incorrect classification.

The learning process of the tagger is described in Figure 1. The initial tagger developed in this toolkit is based on the lexicon which is generated from the golden-standard corpus. To deal with unknown words, the initial tagger utilizes several regular expressions or heuristics whereas the most frequent tag in the training corpus is exploited to label unknown words. The initialized corpus is returned by performing the initial tagger on the raw corpus. By comparing the initialized corpus with the golden one, an object-driven dictionary of pairs (*Object*, *correctTag*) is produced in which *Object* captures the 5-word window context covering the current word and its tag from the initialized corpus, and the *correctTag* is the corresponding tag of the current word in the golden corpus.

There are 27 rule templates applied for Rule selector to select the most suitable rules to build the SCRDR tree. The templates are presented in Table 4. The SCRDR tree of rules is initialized by building the default rule and all exception rules of the default one in form of *if currentTag* = *“**TAG**” then tag* = *“**TAG**”* at the layer-1 exception structure. The

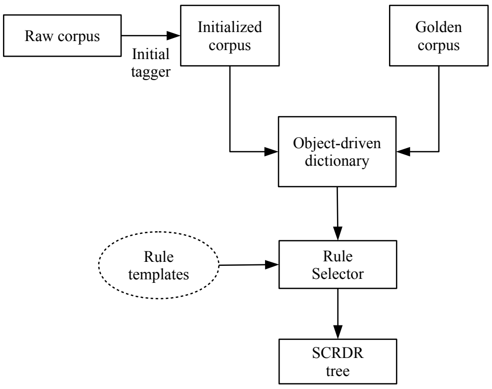


Figure 1. The diagram of the learning process of the RDRPOSTagger learner.

learner then generates new exception rules to every node of the tree due to three constraints described in [14].

Table 4. Short descriptions of rule templates used for Rule selector of RDRPOSTagger

No.	Type	Template
1	Word	$w_{\{-2,-1,0,1,2\}}$
2	Word bigrams	$(w_{-2}, w_0), (w_{-1}, w_0),$ $(w_{-1}, w_1), (w_0, w_1), (w_0, w_2)$
3	Word trigrams	$(w_{-2}, w_{-1}, w_0), (w_{-1}, w_0, w_1),$ (w_0, w_1, w_2)
4	POS tags	$t_{\{-2,-1,0,1,2\}}$
5	POS bigrams	$(t_{-2}, t_{-1}), (t_{-1}, t_1), (t_1, t_2)$
6	Combined	$(t_{-1}, w_0), (w_0, t_1), (t_{-1}, w_0, t_1),$ $(t_{-2}, t_{-1}, w_0), (w_0, t_1, t_2)$
7	Suffix	suffixes of length 1 to 4 of w_0

The tagging process of this tagger firstly assigns tags for unlabeled text by using the initial tagger. Next, for each initially tagged word, the corresponding *Object* will be created. Finally, each word will be tagged by passing its object through the learned SCRDR tree. If the default node is the last fired node satisfying the object, the final tag returned is the tag produced by the initial tagger.

3.2. POS taggers for common languages

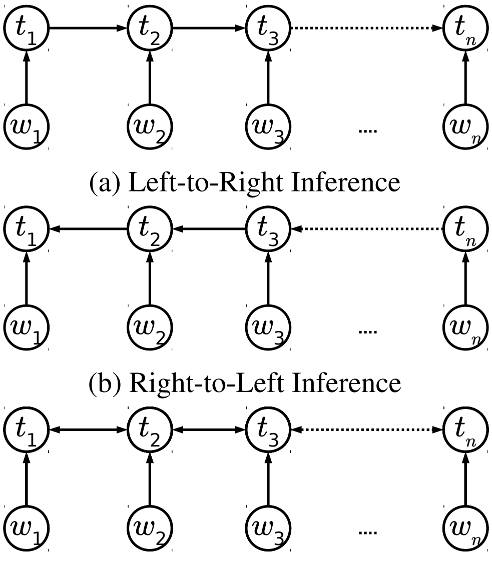
3.2.1. Stanford POS Tagger

Stanford POS Tagger [5] is also a Java-implemented tagger based on stochastic approach. This tagger is the implementation of a log-linear part-of-speech tagging algorithm described in [5] and is developed by Manning and partners at Stanford University. The toolkit is an open-source software[5]. Currently, Stanford POS Tagger has pre-trained models for English, Chinese, Arabic, French and Germany. It can be re-trained in any other language.

The approach described in [5] is based on two main factors, a cyclic dependency network and the MaxEnt model. General idea of the cyclic (or bidirectional) dependency network is to overcome weaknesses of the unidirectional case. In the unidirectional case, only one direction of the tagging sequence is considered at each local point. For instance, in a left-to-right first-order HMM, the current tag t_0 is predicted based on only the previous tag t_{-1} and the current word w_0. However, it is clear that the identity of a tag is also correlated with tag and word identities in both left and right sides. The approach of Stanford POS Tagger follows this idea combined with Maximum Entropy models to provide efficient bidirectional inference.

As reported in [5], with many rich bidirectional-context features and a few additional handcrafted features for unknown words, Stanford POS Tagger can reach the overall accuracy of 97.24% and unknown word accuracy of 89.04%.

[5] http://nlp.stanford.edu/software/tagger.shtml



(a) Left-to-Right Inference

(b) Right-to-Left Inference

(c) Bidirectional Dependency Network

Figure 2. Dependency networks.

3.2.2. ClearNLP

ClearNLP [4] is a toolkit written in Java that contains low-level NPL components (e.g., dependency parsing, named entity recognition, sentiment analysis, part-of-speech tagging), developed by NLP Research Group[6] at Emory University. In our experiments, we use the last released version of ClearNLP – version 3.2.0.

The POS tagging component in ClearNLP is a implementation of the method described in [4]. General idea of this method is to have two models in the tagger and find the most suitable model to assign tags for input sentence based on its domain. Firstly, two separated models, one is optimized for a general domain and the other is optimized for a domain specific to the training data, are trained. They suppose that the domain-specific and generalized models perform better to sentences similar and not similar to the training data, respectively. Hence, during decoding, they dynamically select one of the models by measuring similarities between input sentences and the training data. Some first versions of ClearNLP use dynamic model selection but later versions only use the generalized model to perform the tagging process.

ClearNLP utilizes Liblinear L2-regularization, L1-loss support vector classification [15] for training models and tagging process. It is reported in [4] that this method can gain the overall accuracy of 97.46% for English POS tagging.

4. Experiments

In this section, the process to fix errors in POS-tagged sentences of Vietnamese Treebank corpus is firstly represented. Next, the experimental results of the taggers will be presented.

4.1. Data processing

Vietnamese Treebank corpus was built manually. Some serious errors in this data were found while doing experiments. All of those errors are reported in Table 5.

The #1 row in Table 5 presents error in which the word *"VN"* (the abbreviation of *"Việt Nam"*) is tagged as *Np* (proper name). The right tag for the word *"VN"* in this case is actually *Ny* (abbreviated noun).

The second most frequent error is shown in the #2 row in Table 5. The context is that a number (tagged with *M*) is followed by the word *"tuổi"* (*"years old"*) and the POS tags of *"tuổi"* are not uniform in the whole corpus. There are 184 times in which the tagged sequence is *"<number>/M tuổi/Nu"* (*Nu* is unit noun tag which can be used for *"kilograms"*, *"meters"*, etc.)

[6] http://nlp.mathcs.emory.edu

Table 5. Error analysis on Vietnamese Treebank

Kind of error	Modification	Occurrence
VN/Np	*VN/Ny*	238
<number>/M tuổi/Nu	*<number>/M tuổi/N*	184
Word segmentation error (two underscores between a pair of syllables)	Remove one underscore	105
Tokenization error (two punctuation marks inside a token)	Separate those tokens	99
ð (Icelandic character)	đ (Vietnamese character)	73
More than two tags in one word	Remove the wrong tag	50

and 246 times that the tagged sequence is *"<number>/M tuổi/N"* (*N* is noun). Since the tag *N* is more suitable for the word *"tuổi"* in this situation, all 184 occurrences of *"<number>/M tuổi/Nu"* are replaced by the other one.

There are 105 times of word segmentation error in which the separator of syllables is duplicated. Moreover, there are also 99 times of tokenization error, and 73 times that the character *"đ"* is typed wrongly. The last kind of error is that a single word has two POS tags, which happens 50 times.

Obviously, those errors do affect performance of POS taggers significantly. All of them were discovered during the experiments and were fixed manually to improve the accuracy of the taggers.

After modifying the corpus, we divide it into ten equal partitions which will be used for 10-fold cross-validation. In each fold, nine of ten partitions are used as the training data, the other one is used as the test set. There are about 1.5% – 2% of words in the test set which are unknown in every fold, as shown in Table 6.

Table 6. The experimental datasets

Fold	Total number of words	Number of unknown words
1	63277	1164
2	63855	1203
3	63482	1247
4	62228	1168
5	59854	1056
6	63652	1216
7	63759	1146
8	63071	1224
9	65121	1242
10	63552	1288

4.2. Evaluation

In our experiments, we firstly evaluate the current Vietnamese POS taggers which are vnTagger, JVnTagger and RDRPOSTagger with their default settings. Next, we design a set of features to evaluate the statistical taggers, including two international ones, Stanford Tagger and ClearNLP, and a current Vietnamese one, JVnTagger. There are two terms of the taggers that we measure, which are tagging accuracy and speed. The accuracy is measured using 10-fold cross-validation method on the datasets described above. The speed test is processed on a personal computer with 4 Intel Core i5-3337U CPUs @ 1.80GHz and 6GB of memory. The data used for the speed test is a corpus of 10k sentences collected from Vietnamese websites. This corpus was automatically segmented by UETsegmenter[7] and contains about 250k words. All taggers use their single-threaded implementation for the speed test. Moreover, the test is processed many times to take the average speed of the taggers. We only use the Java-implemented version of RDRPOSTagger in the experiments because it is claimed by the author that this version runs significantly faster than the other one.

We present the performance of the current Vietnamese taggers in Table 7. As we can see, the accuracy results of the taggers are pretty similar to each other's with their default feature sets. The most accurate ones are vnTagger and MaxEnt model of JVnTagger. Especially, these two taggers provide very high accuracies for unknown words. Their specialized features for this kind of word seem to be very effective. Inside the JVnTagger toolkit, the two models provides different results. The MaxEnt model of JVnTagger is far more accurate than the CRFs one. Because these two models use the same feature set, we suspect that the MaxEnt model is more efficient than the CRFs one for Vietnamese POS tagging in term of the tagging accuracy. These two models can provide nearly similar tagging speeds which are 50k and 47k words per second. That may be caused by their same feature set (the CRFs model only has an extra feature so its speed is slightly lower). vnTagger has some complicated features such as the conjunction of two tags and uses an outdated version of Stanford Tagger so that its tagging speed is quite low. Meanwhile, the only tagger that does not make use of statistical approach, RDRPOSTagger, produces an impressive tagging speed at 161k words per second. The tagging speed of a transformation-based tagger is mainly based on the speed of its initial tagger. RDRPOSTagger only uses a lexicon for the initial tagger so that it can perform really fast. Nevertheless, its accuracy for unknown words is not good. Its initial tagger just uses some rules to assign initial tags and then it traverses through the rule tree to determine the final result for the each word. Those rules seem to be unable to handle the unknown words well.

The major of the taggers in our experiments is statistical taggers. In the next evaluation, we will create a unique scheme to evaluate these taggers which are Stanford POS Tagger, ClearNLP and JVnTagger. Although vnTagger is also an statistical one, we do not carry it to the second evaluation because it is based on the basis of Stanford Tagger as mentioned.

It is worth repeating that the performance of each statistical tagger is mainly based on its feature set. The feature set we designed for the second evalution is presented in Table 8. Firstly, a simple feature set will be applied to all of the taggers. This set only contains the 1-gram, 2-gram features for words and some simple one to catch the word shape and the position of the word in the sentence. Next, we will continuously add more advanced features to the feature set to discover which one makes big impact. The three kinds of avanced feature are bidirectional-context, affix and distributional semantic ones. Whereas, the first and the third one are new to the

[7] https://github.com/phongnt570/UETsegmenter

Table 7. The accuracy results (%) of current Vietnamese POS taggers with their default settings. **Ovr.**: the overall accuracy. **Unk.**: the unknown words accuracy. **Spd.**: the tagging speed (words per second)

Feature set	**vnTagger**			**JVn – MaxEnt**			**RDRPOSTagger**			**JVn – CRFs**		
	Accuracy		Spd.	Accuracy		Spd.	Accuracy		Spd.	Accuracy		Spd.
	Ovr.	Unk.		Ovr.	Unk.		Ovr.	Unk.		Ovr.	Unk.	
default	93.88	77.70	13k	93.83	79.60	50k	93.68	66.07	161k	93.59	69.51	47k

Table 8. Feature set designed for experiments of four statistical taggers. **Dist. Semantics**: distributional semantics, ds_i is the cluster id of the word w_i in the Brown cluster set

Feature set	**Template**
Simple	$w_{\{-2,-1,0,1,2\}}$ $(w_{-1}, w_0), (w_0, w_1), (w_{-1}, w_1)$ w_0 has initial uppercase letter? w_0 contains number(s)? w_0 contains punctuation mark(s)? w_0 contains all uppercase letters? w_0 is first or middle or last token?
Bidirectional	$(w_0, t_{-1}), (w_0, t_1)$
Affix	the first syllable of w_0 the last syllable of w_0
Dist. Semantics	ds_{-1}, ds_0, ds_1

current Vietnamese POS taggers. The second one is important for predicting the tags of unknown words.

The performance of four statistical taggers are presented in Table 9. Because JVnTagger does not support the bidirectional-context features so we do not have results for it with the feature sets containing this kind of feature. From Table 9, we can see that with the same simple feature set, these taggers can perform with very similar speeds which are approximately 100k words per second. However, their accuracies are different. With the same feature set, the MaxEnt model of Stanford Tagger can significantly outperform the MaxEnt model of JVnTagger. We suspect that it is caused by the algorithm for optimization and some advanced techniques used in Stanford Tagger. Moreover, with this simple feature set, Stanford Tagger also outperforms any other Vietnamese tagger with its default settings in the first evaluation presented above. Stanford Tagger's techniques seem to be really efficient. Next, inside JVnTagger, with the same feature set, the MaxEnt model still performs better than the CRFs one, again, just like the results conducted in Table 7.

Bidirectional tagging is one of the techniques that have not been applied for current statistical Vietnamese POS taggers. In this experiment, we add two bidirectional-context features which are (w_0, t_{-1}) and (w_0, t_1) to the feature set. These two features capture the information of the tags nearby the current word. The results in Table 9 reveals that bidirectional-context features help to increase the overall accuracy of Stanford Tagger significantly. Moreover, it also draws the tagging speed of this tagger dramatically. However, this kind of feature only makes small impact for ClearNLP which use SVMs for machine learning process, in terms of tagging accuracy and speed.

Table 9. The accuracy results (%) of the four statistical taggers. **spl**: the simple feature set. **bi**: the bidirectional-context feature set. **affix**: the affix features. **ds**: the distributional semantic features

Feature set	**Stanford**			**ClearNLP**			**JVn – MaxEnt**			**JVn – CRFs**		
	Accuracy		Spd.	Accuracy		Spd.	Accuracy		Spd.	Accuracy		Spd.
	Ovr.	Unk.		Ovr.	Unk.		Ovr.	Unk.		Ovr.	Unk.	
spl	93.96	72.19	105k	92.95	68.36	107k	92.53	67.38	102k	91.57	67.34	99k
spl+bi	94.24	72.40	11k	93.08	68.35	93k	N/A					
spl+bi+affix	94.42	78.03	10k	93.83	75.89	90k						
spl+bi+affix+ds	94.53	81.00	8k	94.19	79.01	64k						

Bidirectional-context features do not affect the accuracy of unknown words. Meanwhile, affix feature plays an important role to predict Vietnamese part-of-speech tags. In the next phase of the evaluation, we add the features to catch the first and the last syllable of the current predicting word to discover its impact on the tagging accuracy. As revealed in Table 9, we can conclude that affix features can help to increase the unknown words accuracy sharply, approximately 6% for both Stanford Tagger and ClearNLP. Especially, those features make a very big improvement in the overall accuracy of ClearNLP. Moreover, the tagging speeds of these taggers are affected a little bit with these added features.

The last kind of advanced feature is the distributional semantic one. This is a new technique which has been applied to other languages successfully. To extract this feature, we build 1000 clusters of words based on Brown clustering algorithm [16] using Liang's implementation[8]. The input corpus consists of 2m articles collected from Vietnamese websites. The result in Table 9 shows that distributional semantic features also help to improve the unknown words accuracy of the taggers, at approximately 3% for both taggers. The overall precision is also increased especially in ClearNLP. The tagging speeds of the tagger are decreased about 20% to 30% after adding this kind of feature.

Overall, Stanford POS Tagger is the one that has the best performance with every feature set. ClearNLP also has a good performance. With the full set of features (spl+bi+affix+ds), both of these two international taggers can outperform the current Vietnamese ones with their default settings in term of tagging accuracy. It leads to the fact that some of the specialized features in current Vietnamese taggers are not really useful. The final results of Stanford Tagger and ClearNLP are also the most accurate ones for Vietnamese POS tagging known to us.

5. Conclusion

In this paper, we present an experimental investigation of five part-of-speech taggers for Vietnamese. In the investigation, there are four statistical taggers, Stanford POS Tagger, ClearNLP, vnTagger and JVnTagger. The other one is RDRPOSTagger,

[8] https://github.com/percyliang/brown-cluster

a transformation-based tagger. In term of tagging accuracy, we evaluate the statistical taggers by continuously adding several kinds of feature to them. The result reveals that bidirectional tagging algorithm, affix features and distributional semantic features help to improve the tagging accuracy of the statistical taggers significantly. With the full provided feature set, both Stanford Tagger and ClearNLP can outperform the current Vietnamese taggers. In the speed test, RDRPOSTagger produces an impressive tagging speed. The experimental results also show that tagging speed of any statistical tagger is mainly based on its feature set. With a simple feature set, all of the statistical taggers in our experiments can perform at nearly similar speeds. However, giving an complex feature set to the taggers can draw their tagging speeds deeply.

Acknowledgments

This work has been supported by Vietnam National University, Hanoi (VNU), under Project No. QG.14.04.

References

[1] P. Le-Hong, A. Roussanaly, T. M. H. Nguyen, M. Rossignol, An empirical study of maximum entropy approach for part-of-speech tagging of Vietnamese texts, in: Traitement Automatique des Langues Naturelles-TALN 2010, 2010, p. 12.

[2] D. Q. Nguyen, D. Q. Nguyen, D. D. Pham, S. B. Pham, RDRPOSTagger: A Ripple Down Rules-based Part-Of-Speech Tagger, in: Proceedings of the Demonstrations at the 14th Conference of the European Chapter of the Association for Computational Linguistics, Association for Computational Linguistics, Gothenburg, Sweden, 2014, pp. 17–20.

[3] C.-T. Nguyen, X.-H. Phan, T.-T. Nguyen, JVnTextPro: A tool to process Vietnamese texts, Tech. rep., Tech. rep., Version 2.0, http://jvntextpro. sourceforge. net (2010).

[4] J. D. Choi, M. Palmer, Fast and robust part-of-speech tagging using dynamic model selection, in: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2, Association for Computational Linguistics, 2012, pp. 363–367.

[5] K. Toutanova, D. Klein, C. D. Manning, Y. Singer, Feature-rich part-of-speech tagging with a cyclic dependency network, in: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1, Association for Computational Linguistics, 2003, pp. 173–180.

[6] P.-T. Nguyen, X.-L. Vu, T.-M.-H. Nguyen, V.-H. Nguyen, H.-P. Le, Building a large syntactically-annotated corpus of Vietnamese, in: Proceedings of the Third Linguistic Annotation Workshop, Association for Computational Linguistics, 2009, pp. 182–185.

[7] Z. Huang, W. Xu, K. Yu, Bidirectional LSTM-CRF models for sequence tagging, arXiv preprint arXiv:1508.01991.

[8] E. Brill, A simple rule-based part of speech tagger, in: Proceedings of the workshop on Speech and Natural Language, Association for Computational Linguistics, 1992, pp. 112–116.

[9] R. Prins, G. Van Noord, Unsupervised POS-Tagging Improves Parsing Accuracy and Parsing Efficiency, in: IWPT, 2001.

[10] E. Brill, Unsupervised learning of disambiguation rules for part of speech tagging, in: Proceedings of the third workshop on very large corpora, Vol. 30, Somerset, New Jersey: Association for Computational Linguistics, 1995, pp. 1–13.

[11] O. T. Tran, C. A. Le, T. Q. Ha, Q. H. Le, An experimental study on Vietnamese POS tagging, in: International Conference on Asian Language Processing, 2009. IALP'09., IEEE, 2009, pp. 23–27.

[12] X.-H. Phan, L.-M. Nguyen, Flexcrfs: Flexible conditional random field toolkit.

[13] M. P. Marcus, M. A. Marcinkiewicz, B. Santorini, Building a large annotated corpus of English: The Penn Treebank, Computational linguistics 19 (2) (1993) 313–330.

[14] D. Q. Nguyen, D. Q. Nguyen, D. D. Pham, S. B. Pham, A robust transformation-based learning approach using ripple down rules for part-of-speech tagging, AI Communications (Preprint) (2014) 1–14.

[15] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, C.-J. Lin, LIBLINEAR: A library for large linear classification, The Journal of Machine Learning Research 9 (2008) 1871–1874.

[16] P. F. Brown, P. V. deSouza, R. L. Mercer, V. J. D. Pietra, J. C. Lai, Class-based n-gram models of natural language, Comput. Linguist. 18 (4) (1992) 467–479.