Utilizing State-of-the-art Parsers to Diagnose Problems in Treebank Annotation for a Less Resourced Language

Quy T. Nguyen
University of Information
Technology, Ho Chi Minh City
quynt@uit.edu.vn

Ngan L.T. Nguyen
National Institute
of Informatics, Tokyo
ngan@nii.ac.jp

Yusuke Miyao
National Institute
of Informatics, Tokyo
yusuke@nii.ac.jp

Abstract

The recent success of statistical parsing methods has made treebanks become important resources for building good parsers. However, constructing high-quality annotated treebanks is a challenging task. We utilized two publicly available parsers, Berkeley and MST parsers, for feedback on improving the quality of part-of-speech tagging for the Vietnamese Treebank. Analysis of the treebank and parsing errors revealed how problems with the Vietnamese Treebank influenced the parsing results and real difficulties of Vietnamese parsing that required further improvements to existing parsing technologies.

1 Introduction

Treebanks, corpora annotated with syntactic structures, have become more and more important for language processing. The Vietnamese Treebank (VTB) has been built as part of the national project “Vietnamese language and speech processing (VLSP)” to strengthen automatic processing of the Vietnamese language (Nguyen et al., 2009). However, when we trained the Berkeley parser (Petrov et al., 2006) in our preliminary experiment with VTB and evaluated it using the corpus, the parser only achieved an F-score of 72.1%. This percentage was far lower than the state-of-the-art performance reported for the Berkeley parser on the English Penn Treebank of 90.2% (Petrov et al., 2006). There are two possible reasons for this. First, the quality of VTB is not good enough to construct a good parser that included the quality of the annotation scheme, the annotation guidelines, and the annotation process. Second, parsing Vietnamese is a difficult problem on its own, and we need to seek new solutions to this.

Nguyen et al. (2012) proposed methods of improving the annotations of word segmentation (WS) for VTB. They also evaluated different WS criteria in two applications, i.e., machine translation and text classification. This paper focuses on improving the quality of parts-of-speech (POS) annotations by using state-of-the-art parsers to provide feedback for this process.

The difficulties with Vietnamese POS tagging have been recognized by many researchers (Nghiem et al., 2008; Le et al., 2010). There is little consensus as to the methodology for classifying words. Polysemous words, words with the same surface form but having different meanings and grammar functions, are very popular in the Vietnamese language. For example, the word *“cổ”* can be a noun that means *neck/she*, or an adjective that means *ancient* depending on the context. This characteristic makes it difficult to tag POSs for Vietnamese, both manually and automatically.

The rest of this paper is organized as follows: a brief introduction to VTB and its annotation schemes are provided in Section 2. Then, previous work is summarized in Section 3. Section 4 describes our methods of detecting and correcting inconsistencies in POSs in the VTB corpus. Evaluations of these methods are described in Section 5. Finally, Section 6 explains our evaluations of the Berkeley parser and Minimum-Spanning Tree (MST) parser on different versions of the VTB corpus, which were created by using detected inconsistencies. These results from evaluations are considered to be a way of measuring the effect of automatically detected and corrected inconsistencies. We could observe difficulties with Vietnamese that affected the quality of parsers by analyzing the results from parsing.

Our experiences in using state-of-the-art parsers for treebank annotation, which are presented in this paper, should not only benefit the Vietnamese language, but also other languages with similar

Proceedings of the 7th Linguistic Annotation Workshop & Interoperability with Discourse, pages 19–27,
Sofia, Bulgaria, August 8-9, 2013. ©2013 Association for Computational Linguistics

Label	Name	Example
N	Common noun	*nhân dân {people}*
Np	Proper noun	*Việt Nam {Vietnam}*
Nc	Classifer noun	*con, cái, bức {*}*
Nu	Unit noun	*mét {meter}*
V	Verb	*ngồi {sit}*
A	Adjective	*tốt {good}*
P	Pronoun	*tôi {I}, hắn {he}*
L	Determiner	*mỗi {every}, những {*}*
M	Number	*một {one}*
R	Adverb	*đã, sẽ, đang {*}*
E	Preposition	*trên {on}*
C	Conjunction	*tuy nhiên {however}*
I	Exclamation	*ôi, chao, a ha {*}*
T	Particle	*ạ, ấy, chăng {*}*
B	Foreign word	*internet, email*
Y	Abbreviation	*APEC, WTO, HIV*
S	Affix	*bất, vô, đa {*}*
X	Other	

Table 1: VTB part-of-speech tag set

characteristics.

2 Brief introduction to VTB

The VTB corpus contains 10.433 sentences (274.266 tokens), semi-manually annotated with three layers of WS, POS tagging, and bracketing. The first annotation is produced for each annotation layer by using automatic tools. Then, the annotators revise these data. The WS and POS annotation schemes were introduced by Nguyen et al. (2012). This section briefly introduces POS tag set and a bracketing annotation scheme.

VTB specifies the 18 different POS tags summarized in Table 1 (Nguyen et al., 2010a). Each unit in this table goes with several example words. English translations of these words are included in braces. However, as we could not find any appropriate English translations for some words, these empty translations have been denoted by asterisks (*).

The VTB corpus is annotated with three syntactic tag types: constituency tags, functional tags, and null-element tags. There are 18 constituency tags in VTB. The functional tags are used to enrich information for syntactic trees, such as where functional tag "SUB" is combined with constituency tag "NP", which is presented as "NP-SUB" to indicate this noun phrase is a subject. There are 17 functional tags in VTB. The head word of a phrase is annotated with functional tag "H".

The phrase structures of Vietnamese include three positions: <*pre-head*>, <*head*>, and <*post-head*> (Vietnamese grammar, 1983; Nguyen et al., 2010c). The head word of the phrase is in the <head> position. The words that are in the <pre-head> and <post-head> positions are modifiers of the head word.

There is a special type of noun in Vietnamese that we have called Nc-noun in this paper. Nc-nouns can be classifier nouns or common nouns depending on their modifiers. For example, the Nc-noun *"con"* is a classifier noun if its modifier is the word *"cá {fish}"* (*"con cá"*, which means a specific fish, similar to *"the fish"* in English). However, the Nc-noun *"con {child}"* is a common noun if its modifier is the word *"ghẻ"* (*"con ghẻ"*, which means *"stepchild"* in English). We found that Nc-nouns always appeared in the head positions of noun phrases by investigating the VTB corpus. There is currently little consensus as to the methodology for annotating Nc-nouns (Hoang, 1998; Nguyen et al., 2010b; Nguyen et al., 2010a).

3 Summarization of previous work

Nguyen et al. (2012) described methods of detecting and correcting WS inconsistencies in the VTB corpus. These methods focused on two types of WS inconsistency, variation and structural inconsistency, which are defined below.

Variation inconsistency: is a sequence of tokens that has more than one way of being segmented in the corpus.

Structural inconsistency: occurs when different sequences have similar structures, and thus should be split in the same way, but are segmented in different ways in the corpus. Nguyen et al. (2012) pointed out three typical cases of structural inconsistency that were analyzed as classifier nouns (Nc), affixes (S), and special characters.

Nguyen et al. (2012) analyzed N-gram sequences and phrase structures to detect WS inconsistencies. Then, the detected WS inconsistencies were classified into several patterns of inconsistencies, parts of which were manually fixed to improve the quality of the corpus. The rest were used to create different versions of the VTB corpus. These data sets were evaluated on automatic WS and its applications to text classification and English-Vietnamese statistical machine translations to find appropriate criteria for automatic WS and its applications.

Their experiments revealed that the VAR_FREQ data set achieved excellent results in these applications. The VAR_FREQ data set was the original VTB corpus with manually corrected structural inconsistencies in special characters and selected segmentations with higher frequencies in all detected variations. Therefore, we used the VAR_FREQ data set in our experiments.

4 Methods of detecting and correcting inconsistencies in POS annotations

We propose two kinds of methods of detecting and correcting inconsistencies. They correspond to two different types of POS inconsistency that we call multi-POS inconsistency (MI) and Nc inconsistency (NcI), which are defined as follows.

Multi-POS inconsistency: is a word that is not Nc-noun and has more than one POS tag at each position in each phrase category.

Nc inconsistency: is a sequence of Nc-noun and modifier, in which Nc-noun has more than one way of POS annotation in the VTB corpus.

We separated the POS inconsistencies into these two types of inconsistencies because Nc-nouns are special types of words in Vietnamese. The methods of detecting and correcting NcIs were language-specific methods developed based on the characteristics of Vietnamese. However, as the methods for MIs are rather general, they can be applied to other languages.

4.1 General method for multi-POS inconsistencies

Detection method (MI_DM)

Our main problem was to distinguish MIs from polysemous words, since polysemous words should not be considered inconsistent annotations. Our method was based on the position of words in phrases and phrase categories. This idea resulted from the observation that polysemous words have many POS tags; however, each word usually has only one true POS tag at each position in each phrase category. For example, when a phrase category is a verb phrase, the word *"can"* in the pre-head position of the verb phrase *"(VP (MD can) (VB can))"* should be a modal, but the word *"can"* in the head position should be a verb. Further, the word *"cut"* in the head position of a noun phrase *"(NP (DT a) (JJ further) (NN cut))"* should be a noun, but the word *"cut"* in the head position of the verb phrase *"(VP (VB cut) (NP (NNS costs)))"* should be a verb. This may be more frequent in Vietnamese because it is not an inflectional language i.e., the word form does not change according to tenses, word categories (e.g., nouns, verbs, and adjectives), or number (singular and plural).

The method involved three steps. First, we extracted words in the same position for each phrase category. Second, we counted the number of different POS tags of each word. Words that had more than one POS tag were determined to be multi-POS inconsistencies. For example, in the following two preposition phrases, *"(PP (E-H của) (P chúng_tôi*[1]*)) {of us}"* and *"(PP (C-H của) (P hội_nghị)) {of conference}"*, the words *"của {of}"* appear at the head positions of both phrases, but they are annotated with different POS tags, preposition (E) and conjunction (C). Therefore, they are MIs according to our method.

It should be noted that this method was applied to words that were direct children of a phrase. Embedded phrases, such as *"(PP (E của) (P chúng_tôi))"* in *"(NP (M hai) (Nc-H con) (N mèo) (PP (E của) (P chúng_tôi))) {our two cats}"*, were considered separately.

Correction method (MI_CM)

A multi-POS inconsistency detected with the MI_DM method is denoted by *"w|P1-f1|P2-f2|...|Pn-fn AC"*, where *Pi* (i = 1, 2, ..., n) is a POS tag of word *w*, *fi* is the frequency of POS tag *Pi*, and AC is applying condition of *w*. Our method of correcting the POS tag for POS inconsistency *"w|P1-f1|P2-f2|...|Pn-fn AC"* involves two steps. First, we select the POS tag with the highest frequency of all POS tags of *"w|P1-f1|P2-f2|...|Pn-fn AC"* (*Pmax*). Second, we replace POS tags *Pi* of all instances *(w|Pi)* satisfying condition *AC* with POS tag *Pmax*. For MIs, the AC of word *w* is its phrase category and position in the phrase.

For example, *"toàn bộ|L-27|P-2"* is a multi-POS inconsistency in the pre-head position of a noun phrase. The frequency of POS tag "L" is 27 and the frequency of POS tag "P" is 2. Therefore, "L" is the POS tag that was selected by the MI_CM method. We replace all POS tags *Pi* of instances *"toàn bộ|Pi"* in the pre-head positions of noun phrases with POS tag "L".

4.2 Language-specific method for classifier nouns

Detection method

As mentioned in Section 2, an Nc-noun can be

[1] We used underscore "_" to link syllables of Vietnamese compound words.

annotated with POS tag "Nc" or "N" depending on the modifier that follows that Nc-noun. Analyzing the VTB corpus revealed that Nc-nouns had two characteristics. First, an Nc-noun that is followed by the same word at each occurrence is usually annotated with the same POS tag. Second, an Nc-noun that is followed by a phrase or nothing at each occurrence is annotated with the same POS tag. Based on these two cases, we propose two methods of detecting NcIs, which we have called NcI_DM1 and NcI_DM2. They are described below.

NcI_DM1: We counted Nc-nouns in VTB that had two or more ways of POS annotation, satisfying the condition that Nc-nouns are followed by a phrase or nothing. For example, the Nc-noun *"con"* in *"(NP (M 2) (N-H con)) {2 children}"* is followed by nothing or it is followed by a prepositional phrase as in *"(NP (L các) (N-H con) (PP (E-H của) (P tôi))) {my children}"*.

NcI_DM2: We counted two-gram sequences beginning with an Nc-noun in VTB that had two or more ways of POS annotation of the Nc-noun, satisfying the conditions that two tokens were all in the same phrase and and they all had the same depth in a phrase. For example, the Nc-noun *"con"* in the two-gram *"con gái {daughter}"* was sometimes annotated "Nc", and sometimes annotated "N" in VTB; in addition, as *"con"* and *"gái"* in the structure *"(NP (Nc-H con) (N gái) (PP (E-H của) (P tôi))) {my daughter}"* were in the same phrase and have the same depth, *"con"* was an NcI.

Correction method

We denoted NcIs with *"w|P1-f1|P2-f2|...|Pn-fn AC"* similarly to MIs. We also replaced the POS tag of Nc-nouns with the highest frequency tag. The only differences were the applying conditions that varied according to the previous two cases of NcIs.

- For Nc inconsistencies detected by the NcI_DM1 method, AC is defined as follows: *w* is an Nc-noun that is followed by nothing or a phrase.

- For Nc inconsistencies detected by the NcI_DM2 method, AC is defined as follows: *w* is an Nc-noun that must be followed by a word, *m*.

5 Results and evaluation

We detected and corrected MIs and NcIs based on the two data sets, ORG and VAR_FREQ. The ORG data set was the original VTB corpus and VAR_FREQ was the original corpus with modifications to WS annotation. This setting was made similar to that used by Nguyen et al. (2012) to enable comparison.

There are a total of 128,871 phrases in the VTB corpus. The top five types of phrases are noun phrases (NPs) (representing 49.6% of the total number of phrases), verb phrases (VPs), prepositional phrases (PPs), adjectival phrases (ADJPs), and quantity phrases (QPs), representing 99.1% of the total number of phrases in the VTB corpus. We analyzed the VTB corpus based on these five types of phrases.

5.1 Results for detected POS inconsistencies

Tables 2 and 3 show the overall statistics for MIs and NcIs for each phrase category. The second and third columns in these tables indicate the numbers of inconsistencies and their instances that were detected in the ORG data set. The fourth and fifth columns indicate the numbers of inconsistencies and their instances that were detected in the VAR_FREQ data set. The rows in Table 3 indicate the number of NcIs and the number of instances detected with the NcI_DM1 and NcI_DM2 methods.

According to Table 2, most of the MIs occurred in noun phrases, representing more than 72% of the total number of MIs. All NcIs in Table 3 are also in noun phrases. There are two possible reasons for this. First, noun phrases represent the majority of phrases in VTB (represent 49.6% of the total number of phrases in the VTB corpus). Second, nouns are sub-divided into many other types (common noun (N), classifier noun (Nc), proper noun (Np), and unit noun (Nu)) (mentioned in Section 2), which may confuse annotators in annotating POS tags for nouns. In addition, the high number of NcIs in Table 3 indicate that it is difficult to distinguish between Nc and other types of nouns. Therefore, we need to have clearer annotation guidelines for this.

5.2 Evaluation of methods to detect and correct inconsistencies

We estimated the accuracy of our methods which detected and corrected inconsistencies in POS tag-

Phrase	ORG		VAR_FREQ	
	Inc	Ins	Inc	Ins
NP	792	28,423	752	27,067
VP	221	10,158	139	10,110
ADJP	64	1,302	61	1,257
QP	4	22	4	22
PP	14	5,649	13	5,628
Total	**1,095**	**45,554**	**969**	**44,084**

Table 2: Statistics for multi-POS inconsistencies for each phrase category in VTB. Number of Inconsistencies (Inc) and Number of Instances (Ins).

Detection method	ORG		VAR_FREQ	
	Inc	Ins	Inc	Ins
NcI_DM1	52	3,801	51	3,792
NcI_DM2	338	2,468	326	2,412
Total	***390***	***6,269***	***377***	***6,204***

Table 3: Statistics for Nc inconsistencies in head positions of noun phrases in VTB.

ging by manually inspecting inconsistent annotations. We manually inspected the two data sets of ORG_EVAL and ORG_POS_EVAL. To create ORG_EVAL, we randomly selected 100 sentences which contained instances of POS inconsistencies in the ORG data set. ORG_EVAL contained 459 instances of 157 POS inconsistencies. ORG_POS_EVAL was the ORG_EVAL data set with corrections made to multi-POS inconsistencies and Nc inconsistencies with our methods of correction above.

Detection: We manually checked POS inconsistencies and found that 153 cases out of 157 POS inconsistencies (97.5%) were actual inconsistencies. There were four cases that our method detected as multi-POS inconsistencies, but they were actually ambiguities in Vietnamese POS tagging. They were polysemous words whose meanings and POS tags depended on surrounding words, but did not depend on their positions in phrases. For example, the word *"sáng"* in the post-head positions of the verb phrases VP1 and VP2 below, can be a noun that means *morning* in English, or it can be an adjective that means *bright*, depending on the preceding verb.

VP1: (VP (V-H thắp) (A sáng) {lighten bright}

VP2: (VP (V-H đi) (N sáng) {go in the morning}

ORG_EVAL	ORG_POS_EVAL	No. of Instances
correct	correct	404
incorrect	correct	41
correct	incorrect	11
incorrect	incorrect	3
Total		***459***

Table 4: Comparison of POS tags for 459 instances in ORG_EVAL with those in ORG_POS_EVAL.

PoPOS	Counts	Examples
Nc-N	385	*người {the, person}*
N-V	186	*mất mát {loss}*
N-Np	176	*Hội {association}*
N-A	144	*khó khăn {difficult}*
V-A	92	*phải {must, right}*

Table 5: Top five pairs of confusing POS tags.

Correction: Table 4 shows results of comparison of the POS tags for 459 instances in ORG_EVAL and those in ORG_POS_EVAL. These results indicate that there are instances whose POS tags are incorrect in ORG_EVAL but correct in ORG_POS_EVAL (the third row in Table 4), and there are instances whose POS tags are correct in ORG_EVAL but incorrect in ORG_POS_EVAL (the fourth row in Table 4). The results in Table 4 indicate that, the number of correct POS tags in ORG_POS_EVAL (445 instances, representing 96.9% of the total number of instances) is higher than that in ORG_EVAL (415 instances, representing 90.4% of the total number of instances). This means our methods of correcting inconsistencies in POS tagging improved the quality of treebank annotations.

5.3 Analysis of detected inconsistencies

We analyzed the detected POS inconsistencies to find the reasons for inconsistent POS annotations. We classified the detected POS inconsistencies according to pairs of their POS tags. There were a total of 85 patterns of pairs of POS tags. Table 5 lists the top five confusing patterns (PoPOS), their counts of inconsistencies (Counts), and examples. It also seemed to be extremely confusing for the annotators to distinguish types of nouns (Nc and N, and N and Np) and distinguish nouns from other types of words (such as verbs, adjectives, and pronouns).

We investigated POS inconsistencies and the annotation guidelines (Nguyen et al., 2010b; Nguyen et al., 2010a; Nguyen et al., 2010c) to find why common nouns were sometimes tagged as classifier nouns and vice versa, and verbs were sometimes tagged as common nouns and vice versa, and so on. We found that these POS inconsistencies belonged to polysemous words that were difficult to tag.

The difficulties with tagging polysemous words were due to four main reasons: (1) The POS of a polysemous word changes according to the function of that polysemous word in each phrase category or changes according to the meaning of surrounding words. Although polysemous words are annotated with different POS tags, they do not change their word form. (2) The way polysemous words are tagged according to their context is not completely clear in the POS tagging guidelines. (3) Annotators referred to a dictionary that had been built as part of the VLSP project (Nguyen et al., 2009) (VLSP dictionary) to annotate the VTB corpus. However, this dictionary lacked various words and did not cover all contexts for the words. For example, *"hơn {more than}"* in Vietnamese is an adjective when it is the head word of an adjectival phrase, but *"hơn {over}"* is an adverb when it is the modifier of a quantifier noun (such as *"hơn 200 sinh viên {over 200 students}"*). However, the VLSP dictionary only considered *"hơn"* to be an adjective (*"tôi hơn nó hai tuổi {I am more than him two years old}"*). No cases where *"hơn"* was an adverb were mentioned in this dictionary. (4) There are several overlapping but conflicting instructions across the annotation guidelines for different layers of the treebank. For example, the combinations of affixes and words they modify to create compound words are clear in the WS guidelines, but POS tagging guidelines treat affixes as words and they are annotated as POS tags "S". For words modifying quantifier nouns, such as *"hơn and gần {over and about}"*, the POS tagging guidelines treat them as adjectives, but the bracketing guidelines treat them as adverbs. Therefore, our method detected multi-POS inconsistencies as *"hơn|A-135|R-51"*, *"gần|A-102|R-5"* at the pre-head positions of noun phrases. Since the frequencies of the adjective tags were greater than those of adverb tags (fA > fR), these words were automatically assigned to adjective POS tags (A) according to our method of correction. These were POS inconsistencies that our method of correction could not be applied to, because the frequency of incorrect POS tags was higher than that of actual POS tags.

6 Evaluation of state-of-the-art parsers on VTB

We carried out experiments to evaluate two popular parsers, a syntactic parser and a dependency parser, on different versions of the VTB corpus. Some of these data sets were made the same as the data settings for WS in Nguyen et al. (2012). The other data sets contained changes in POS annotations following our methods of correcting inconsistencies presented in Section 4. We could observe how the problems with WS and POS tagging influenced the quality of Vietnamese parsing by analyzing the parsing results.

6.1 Experimental settings

Data. Nine configurations of the VTB corpus were created as follows:

- ORG: The original VTB corpus.
- BASE, STRUCT_AFFIX, STRUCT_NC, VAR_SPLIT, VAR_COMB, and VAR_FREQ correspond to different settings for WS described in Nguyen et al. (2012).
- ORG_POS: The ORG data set with corrections for multi-POS inconsistencies and Nc inconsistencies by using the methods in Section 4.1 and 4.2.
- VAR_FREQ_POS: The VAR_FREQ data set with corrections for multi-POS inconsistencies and Nc inconsistencies by using the methods in Section 4.1 and 4.2.

Each of the nine data sets was randomly split into two subsets for training and testing our parser models. The training set contained 9,443 sentences, and the testing set contained 1,000 sentences.

Tools

We used the Berkeley parser (Petrov et al., 2006) to evaluate the syntactic parser on VTB. This parser has been used in experiments in English, German, and Chinese and achieved an F1 of 90.2% on the English Penn Treebank.

We used the conversion tool built by Johansson et al. (2007) to convert VTB into dependency trees.

We used the MST parser to evaluate the dependency parsing on VTB. This parser was evaluated on the English Penn Treebank (Mcdonald et al., 2006a) and 13 other languages (Mcdonald et al., 2006b). Its accuracy achieved 90.7% on the English Penn Treebank.

We made use of the bracket scoring program EVALB, which was built by Sekine et al. (1997),

Data sets	Bracketing F-measures
ORG	72.10
BASE	72.20
STRUCT_AFFIX	72.60
STRUCT_NC	71.92
VAR_SPLIT	72.03
VAR_COMB	72.46
VAR_FREQ	72.34
ORG_POS	72.72
VAR_FREQ_POS	**73.21**

Table 6: Bracketing F-measures of Berkeley parser on nine configurations of VTB corpus.

Data set	UA	LA
ORG	50.51	46.14
BASE	53.90	50.14
STRUCT_AFFIX	54.00	50.25
STRUCT_NC	53.88	49.96
VAR_SPLIT	53.95	50.14
VAR_COMB	53.93	50.27
VAR_FREQ	54.21	50.41
ORG_POS	54.20	50.37
VAR_FREQ_POS	**57.87**	**53.19**

Table 7: Dependency accuracy of MSTParser on nine configurations of VTB corpus. Unlabeled Accuracy (UA), Labeled Accuracy (LA).

to evaluate the performance of the Berkeley parser. As an evaluation tool was included in the MST parser tool, we used it to evaluate the MST parser.

6.2 Experimental results

The bracketing F-measures of the Berkeley parser on nine configurations of the VTB corpus are listed in Table 6. The dependency accuracies of the MST parser on nine configurations of the VTB corpus are shown in Table 7. These results indicate that the quality of the treebank strongly affected the quality of the parsers.

According to Table 6, all modifications to WS inconsistencies improved the performance of the Berkeley parser except for STRUCT_NC and VAR_SPLIT. More importantly, the ORG_POS model achieved better results than the ORG model, and the VAR_FREQ_POS model achieved better results than the VAR_FREQ model, which indicates that the modifications to POS inconsistencies improved the performance of the Berkeley parser. The VAR_FREQ_POS model scored 1.11 point higher than ORG, which is a significant improvement.

Dependency accuracies of the MST parser in Table 7 indicate that all modifications to POS inconsistencies improved the performance of the MST parser. All modifications to WS inconsistencies also improved the performance of the MST parser except for STRUCT_NC. The VAR_FREQ_POS model scored 7.36 points higher than ORG, which is a significant improvement.

APSs	CCTs and Freq
A M N	NP-79|ADJP-27
A V	VP-56|ADJP-78|NP-2

Table 8: Examples of ambiguous POS sequences (APSs), their CCTs, and frequency of each CCT (Freq)

6.3 Analysis of parsing results

The results for the Berkeley parser and MST parser trained on the POS-modified versions of VTB were better than those trained on the original VTB corpus, but they were still much lower than the performance of the same parsers on the English language. We analyzed error based on the output data of the best parsing results (VAR_FREQ_POS) for the Berkeley parser, and found that the unmatched annotations between gold and test data were caused by ambiguous POS sequences in the VTB corpus.

An ambiguous POS sequence is a sequence of POS tags that has two or more constituency tags. For example, there are the verb phrase *"(VP (R đang) (A cặm_cụi) (V làm)) {* (be) painstakingly doing}"* and the adjectival phrase *"(ADJP (R rất) (A dễ) (V thực_hiện)) {very easy (to) implement}"* in the training data of VAR_FREQ_POS. As these two phrases have the same POS sequence *"R A V"*, *"R A V"* is an ambiguous POS sequence, and VP and ADJP are confusing constituency tags (CCTs). We found 42,373 occurrences of 213 ambiguous POS sequences (representing 37.02% of all phrases) in the training data of VAR_FREQ_POS. We also found 1,065 occurrences of 13 ambiguous POS sequences in the parsing results for VAR_FREQ_POS. Some examples of ambiguous POS sequences, their CCTs, and the number of occurrences of each CCT in the training data of VAR_FREQ_POS are listed in Table 8.

We classified the detected ambiguous POS sequences according to pairs of different CCTs to find the reasons for ambiguity in each pair. There were a total of 42 pairs of CCTs, whose top three pairs, along with their counts of types of ambiguous POS sequences, and examples of ambigu-

Pairs of CCTs	Counts	Examples
NP-VP	61	P V N, ...
VP-ADJP	54	R A V, A V N, ...
ADJP-NP	52	A M N, ...

Table 9: Top three pairs of confusing constituency tags

Pairs of CCTs	1	2
NP-VP	M, L ,R ,V	N, R, M, P, A
VP-ADJP	A, R	N, R
ADJP-NP	N, R	R, M, A, L

Table 10: Statistics for POS tags at pre-head position of each phrase category.

ous POS sequences are listed in Table 9. We extracted different POS tags at each position of each phrase category for each pair of CCTs, based on the ambiguous POS sequences. For example, the third row in Table 9 has "R A V" and "A V N", which are two ambiguous POS sequences that were sometimes annotated as VP and sometimes annotated as ADJP. The different POS tags that were extracted from the pre-head positions of VPs based on these two POS sequences were "R, A" and "R" was the POS tag that was extracted from the pre-head positions of ADJPs based on these two POS sequences. These POS tags are important clues to finding reasons for ambiguities in POS sequences.

Table 10 summarizes the extracted POS tags at pre-head positions for the top three pairs of CCTs. For example, the POS tags in row NP-VP and column 1 are in the pre-head positions of NP and the POS tags in row NP-VP and column 2 are in the pre-head positions of VP. By comparing these results with the structures of the pre-head positions of phrase categories in VTB bracketing guidelines (Nguyen et al., 2010c), we found many cases that were not annotated according to instructions in the VTB bracketing guidelines, such as those according to Table 10, where an adjective (A) is in the pre-head position of VP, but according to the VTB bracketing guidelines, the structure of the pre-head position of VB only includes adverb (R).

We investigated cases that had not been annotated according to the guidelines, and found two possible reasons that caused ambiguous POS sequences. First, although our methods improved the quality of the VTB corpus, some POS annotation errors remained in the VTB corpus. These POS annotation errors were cases to which our methods could not be applied (mentioned in Section 5). Second, there were ambiguities in POS sequences caused by Vietnamese characteristics, such as the adjectival phrase *"(ADJP (R đang) (N ngày_đêm) (A đau_đớn)) {* day-and-night painful}"* and the noun phrase *"(NP (R cũng) (N sinh_viên) (A giỏi)) {also good student}"* that had the same POS sequence of "R N A".

Therefore, POS annotation errors need to be eliminated from the VTB corpus to further improve its quality and that of the Vietnamese parser. We not only need to eliminate overlapping but conflicting instructions, which were mentioned in Section 5.3, from the guidelines, but we also have to complete annotation instructions for cases that have not been treated (or not been clearly treated) in the guidelines. We may also need to improve POS tag set because adverbs modifying adjectives, verbs and nouns are all presently tagged as "R", which caused ambiguous POS sequences, such as the ambiguous POS sequence "R N A" mentioned above. If we use different POS tags for the adverb *"đang"*, which modifies the adjective *"đau đớn {painful}"*, and the adverb *"cũng"*, which modifies the noun *"sinh viên {student}"*, we can eliminate ambiguous POS sequences in these cases.

7 Conclusion

We proposed several methods of improving the quality of the VTB corpus. Our manual evaluation revealed that our methods improved the quality of the VTB corpus by 6.5% with correct POS tags. Analysis of inconsistencies and the annotation guidelines suggested that: (1) better instructions should be added to the VTB guidelines to help annotators to distinguish difficult POS tags, (2) overlapping but conflicting instructions should be eliminated from the VTB guidelines, and (3) annotations that referred to dictionaries should be avoided.

To the best of our knowledge, this paper is the first report on evaluating state-of-the-art parsers used on the Vietnamese language. The results obtained from evaluating these two parsers were used as feedback to improve the quality of treebank annotations. We also thoroughly analyzed the parsing output, which revealed challenging issues in treebank annotations and in the Vietnamese parsing problem itself.

References

Anna M. D. Sciullo and Edwin Williams. 1987. *On the definition of word*. The MIT Press.

Fei Xia. 2000. *The part-of-speech tagging guidelines for the penn chinese treebank (3.0)*.

Minh Nghiem, Dien Dinh and Mai Nguyen. 2008. *Improving Vietnamese POS tagging by integrating a rich feature set and Support Vector Machines*. Proceedings of RIVF 2008, pages: 128–133.

Phe Hoang. 1998. *Vietnamese Dictionary*. Scientific & Technical Publishing.

Phuong H. Le, Azim Roussanaly, Huyen T. M. Nguyen and Mathias Rossignol. 2010. *An empirical study of maximum entropy approach for part-of-speech tagging of Vietnamese texts*. Proceedings of TALN 2010 Conference. Montreal, Canada.

Quy T. Nguyen, Ngan L.T. Nguyen and Yusuke Miyao. 2012. *Comparing Different Criteria for Vietnamese Word Segmentation*. Proceedings of 3rd Workshop on South and Southeast Asian Natural Language Processing (SANLP), pages: 53–68.

Richard Johansson and Pierre Nugues. 2007. *Extended Constituent-to-dependency Conversion for English*. Proceedings of NODALIDA, Tartu, Estonia, pages: 105–112.

Ryan Mcdonald and Fernando Pereira. 2006a. *Online Learning of Approximate Dependency Parsing Algorithms*. Proceedings of 11th Conference of the European Chapter of the Association for Computational Linguistics: EACL 2006, pages: 81–88.

Ryan Mcdonald, Kevin Lerman and Fernando Pereira. 2006b. *Multilingual Dependency Analysis with a Two-Stage Discriminative Parser*. Proceedings of Tenth Conference on Computational Natural Language Learning (CoNLL-X), Bergan, Norway, pages: 216–220.

Slav Petrov, Leon Barrett, Romain Thibaux and Dan Klein. 2006. *Learning accurate, compact, and interpretable tree annotation*. Proceedings of 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics, pages: 433–440.

Thai P. Nguyen, Luong X. Vu and Huyen T.M. Nguyen. 2010a. *VTB part-of-speech tagging guidelines*.

Thai P. Nguyen, Luong X. Vu and Huyen T.M. Nguyen. 2010b. *VTB word segmentation guidelines*.

Thai P. Nguyen, Luong X. Vu, Huyen T.M. Nguyen, Hiep V. Nguyen and Phuong H. Le. 2009. *Building a large syntactically-annotated corpus of Vietnamese*. Proceedings of Third Linguistic Annotation Workshop, pages: 182–185.

Thai P. Nguyen, Luong X. Vu, Huyen T.M. Nguyen, Thu M. Dao, Ngoc T.M. Dao and Ngan K. Le. 2010c. *VTB bracketing guidelines*.

Vietnamese grammar. 1983. Social Sciences Publishers.