An Empirical Investigation of Error Types in Vietnamese Parsing

Quy T. Nguyen[1,*] **Yusuke Miyao**[2] **Hiroshi Noji**[3] **Nhung T.H. Nguyen**[4]
[1]University of Information Technology, Ho Chi Minh City, Vietnam
[2]National Institute of Informatics, Tokyo, Japan
[3]Artificial Intelligence Research Center, AIST, Tokyo, Japan
[4]University of Science, Ho Chi Minh City, Vietnam
quynt@uit.edu.vn, yusuke@nii.ac.jp
hiroshi.noji@aist.go.jp, nthnhung@fit.hcmus.edu.vn

Abstract

Syntactic parsing plays a crucial role in improving the quality of natural language processing tasks. Although there have been several research projects on syntactic parsing in Vietnamese, the parsing quality has been far inferior than those reported in major languages, such as English and Chinese. In this work, we evaluated representative constituency parsing models on a Vietnamese Treebank to look for the most suitable parsing method for Vietnamese. We then combined the advantages of automatic and manual analysis to investigate errors produced by the experimented parsers and find the reasons for them. Our analysis focused on three possible sources of parsing errors, namely limited training data, part-of-speech (POS) tagging errors, and ambiguous constructions. As a result, we found that the last two sources, which frequently appear in Vietnamese text, significantly attributed to the poor performance of Vietnamese parsing.

Title and Abstract in Vietnamese

Khám phá dựa trên thực nghiệm các kiểu lỗi trong phân tích cú pháp tiếng Việt

Phân tích cú pháp đóng vai trò then chốt trong việc cải thiện chất lượng các hệ thống xử lý ngôn ngữ tự nhiên. Mặc dù đã có một vài dự án nghiên cứu về phân tích cú pháp tiếng Việt, chất lượng của những công trình này thấp hơn nhiều so với các nghiên cứu trên các ngôn ngữ phổ biến như tiếng Anh và tiếng Trung Quốc. Trong bài báo này, chúng tôi đánh giá các phương pháp phân tích cú pháp tiêu biểu trên ngân hàng cây cú pháp tiếng Việt để tìm ra phương pháp phân tích cú pháp phù hợp nhất cho tiếng Việt. Sau đó, chúng tôi kết hợp những ưu điểm của phân tích tự động và phân tích thủ công để khám phá các kiểu lỗi mà những phương pháp phân tích cú pháp được thực nghiệm mắc phải và tìm nguyên nhân cho những lỗi đó. Phân tích của chúng tôi tập trung vào ba nguồn chính gây ra lỗi, cụ thể là dữ liệu huấn luyện ít, lỗi trong việc gán nhãn từ loại, và các cấu trúc nhập nhằng. Kết quả cho thấy rằng lỗi trong việc gán nhãn từ loại và cấu trúc nhập nhằng là hai nguyên nhân chính gây ra hiệu suất thấp trong phân tích cú pháp tiếng Việt.

1 Introduction

Syntactic parsing plays a crucial role in improving the quality of natural language processing (NLP) applications and speech processing. Parsing has been broadly studied for languages such as English and Chinese, which leads to the development of many parsing methods. For example, Klein and Manning (2003), Matsuzaki et al. (2005), and Petrov and Klein (2007) improved probabilistic context-free grammar (PCFG) parsing by enriching contextual information, Finkel et al. (2008) and Hall et al. (2014) employed feature-based conditional random fields (CRFs), Zhu et al. (2013) and Coavoux and Crabbé (2017) employed shift-reduce algorithms, while Durrett and Klein (2015) and Dyer et al. (2016) used neural networks.

The parsing problem has not been studied thoroughly enough for Vietnamese. Since a Vietnamese Treebank was built by Nguyen et al. (2009), a few researchers have adapted available constituent parsers

*This work was carried out while the first author was a PhD student at The Graduate University for Advanced Studies (SOKENDAI), Japan.

Proceedings of the 27th International Conference on Computational Linguistics, pages 3075–3089
Santa Fe, New Mexico, USA, August 20-26, 2018.

to construct a parser for Vietnamese; e.g., AC. Le et al. (2009) modified the Bikel's parser (Bikel, 2004) and Le-Hong et al. (2012) modified the LTAG parser developed by LORIA laboratory[1]. However, the parsing accuracies were far lower than the performances reported in English, Chinese, and French.

It is hard to parse Vietnamese text due to difficulties that result from characteristics of the Vietnamese language[2]. Vietnamese does not have word delimiters and inflectional morphemes in comparison to English. While similar problems also occur in Chinese (Xia et al., 2000), parsing Vietnamese may be more difficult because the modern Vietnamese writing system is based on Latin characters, which represent the pronunciation but not the meanings of words. As a result, there are many polysemous expressions in Vietnamese, i.e., expressions having the same surface form with different interpretations. Difficulties in Vietnamese parsing are also caused by word orders. Although Vietnamese is a subject-verb-object (SVO) language like English and Chinese, its word orders are different from these languages. For example, the word order in Vietnamese noun phrases (*NPs*) is exactly the same as those in simple sentences, which leads to ambiguities in labelling these two types of expressions. In addition, other problems, such as word omission, contribute to many complications in parsing Vietnamese texts.

Another possible reason for difficulties in Vietnamese parsing is treebank design[3]. For example, syntactic categories in the Vietnamese Treebank by Nguyen et al. (2016) are similar to those in the Penn Treebank, e.g., *NP* for noun phrases and *VP* for verb phrases. These categories were not fine-grained enough to distinguish various syntactic constructions in Vietnamese. In addition, there are ambiguous constructions that have the same tag sequences (including POS tags and phrase tags), but are bracketed in different ways. For instance, the sequence of POS tags *Nn Vv* can be bracketed as a noun phrase (*NP*) in which *Vv* is a modifier of the head noun *Nn*. The sequence can also be bracketed as a simple sentence where *Nn* is the subject and *Vv* is the predicate.

A worthwhile effort would be to analyze parsing errors to improve the quality of parsers. Based on this analysis, we can consider how to modify parsing models such that they can capture the necessary syntactic information. Several researchers (Levy and Manning, 2003; Hara et al., 2009) have manually investigated parsing errors. Manual investigations can help us find reasons for parsing errors that are specific to languages. However, this method cannot be applied to a large amount of parsing outputs and is also limited to a few syntactic phenomena. Kummerfeld et al. (2012) developed a tool to automatically classify English parsing errors within a set of predefined error types such as NP attachment and VP attachment. However, based on these analyzing results, we cannot specify any reasons to explain why the error types occurred, which is important clues to improve parsers.

In this paper, we firstly evaluated representative parsing models on the Vietnamese Treebank. We then investigated three possible sources of errors produced by parsers, viz., limited training data, confusing POS tags, and ambiguous constructions. Our analysis method combined an automatic tool and manual analysis. We used Kummerfeld et al. (2012)'s tool in the automatic phase to quantify how often the types of errors occurred in the experimented parsers. By comparing the results obtained from this analysis, we could identify which types of errors could be solved by different parsers as well as which errors were difficult for parsers to address. We manually investigated parsing errors in the second phase based on the results obtained from the analysis in the first phase to find the reasons for each error type.

2 Parsing evaluation

We applied six representative parsers in the literature in English and Chinese to the Vietnamese Treebank. For each parser, we used parameter settings that had the best accuracy in English.

Stanford-Unlex (Klein and Manning, 2003) is an unlexicalized probabilistic context free grammar (PCFG) parser where the coarse categories of PCFG, such as *NP* for noun phrases, are enriched by several simple structural annotations, e.g., markovization before splitting several symbols according to a manual grammar.

Stanford-RNN (Socher et al., 2013) is a combination of a PCFG with a recursive neural network,

[1]http://www.loria.fr

[2]An example illustrating Vietnamese charateristics is presented in Appendix C.

[3]Vietnamese Treebank's POS tags and constituent tags are described in Appendices A and B.

Parsers	Tagging accuracy	R	P	F
Stanf-Unlex	89.92	52.15	56.38	54.18
Stanf-RNN	91.83	63.66	66.15	64.88
Berkeley	93.94	70.38	73.48	71.90
Epic-CRF	93.15	72.20	72.96	72.58
Epic-NeuralCRF	93.85	71.68	72.42	72.05
RNNGs-D	94.65	71.94	72.45	72.19
RNNGs-G	**94.65**	**71.71**	**74.21**	**72.94**

Table 1: PARSEVAL scores on the test set for all parsers in our experiments.

which reranks the outputs of Stanford-Unlex. Phrase representations in this model are learned via a recursive neural network that is conditioned on the coarse PCFG.

Berkeley parser (Petrov et al., 2006) is an unlexicalized PCFG parser where the PCFG is automatically enriched and generalized by using informative latent annotations.

Epic-CRF (Hall et al., 2014) is a CRF parser in which the anchored rules are scored by using sparse features of the surface spans. The base grammar in this parser employs parent annotations, a subset of annotations in Stanford-Unlex.

Epic-NeuralCRF (Durrett and Klein, 2015) is also a CRF parser but the anchored rules are scored using dense features that are computed via a feedforward neural network.

RNNGs (Dyer et al., 2016) is a top-down transition-based neural model in which a recurrent neural network conditions on a full input sentence to parameterize decisions. The RNNGs parser supports two models, i.e., discriminative (RNNGs-D) and generative models (RNNGs-G).

We used the Vietnamese Treebank developed by Nguyen et al. (2016) which includes 10,377 sentences (containing 232,838 words and 280,505 syllables). We randomly selected 1,000 sentences for the development (dev) set, 1,000 sentences for the test set, and the rest, including 8,377 sentences, was used for training. We used the input data with gold word segmentation[4] and predicted POS tags (PredictedPOS)[5].

Table 1 summarizes the PARSEVAL results of each parser on the test set. We can see from this table that RNNGs-G, which is a top-down transition-based neural model, achieved the best accuracy on the Vietnamese Treebank, while the differences among the RNNGs-G, RNNGs-D, Epic-CRF, and Epic-NeuralCRF were not very significant. This indicates that parsing models based on CRF and neural networks are equally good for Vietnamese. However, these parsers still achieved far less accuracy in comparison to that in English. One possible reason is that although each of the parsing models had its own advantages, these were insufficient to address all Vietnamese constructions.

Although Stanford-Unlex and Stanford-RNN also enriched the coarse categories of PCFG with contextual information, they achieved far less accuracy in comparison with the RNNGs parser, Berkeley parser, Epic-CRF, and Epic-NeuralCRF. The main reason for this is that Stanford-Unlex and Stanford-RNN usually split tags into subcategories in response to linguistic trends in the English Penn Treebank. For example, determiners are distinguished from demonstratives, or the POS tag of *IN* is divided into six subcategories. It was difficult to apply these parsers to the Vietnamese Treebank because many linguistic trends occurring in the English Penn Treebank are not found in the Vietnamese Treebank.

In comparison with Stanford-Unlex and Stanford-RNNs, the Berkeley parser, Epic-CRF, and Epic-NeuralCRF obtained far higher accuracies. The reason is that the Berkeley parser, Epic-CRF, and Epic-NeuralCRF used automatic methods to enrich contextual information, which were easy to apply to other languages including Vietnamese. This also indicated that enriching contextual information is beneficial for Vietnamese parsing. Contextual information can be fine-grained categorizations or features extracted from the surface spans of anchor rules[6]. However, it seems that the contextual information extracted on the basis of CFG rules did not sufficiently capture Vietnamese constructions. While RNNGs modeled

[4]We used gold word segmentation for all experiments in this research.

[5]The POS tags were predicted by the parsers. In the case of RNNGs, since the parser cannot predict POS tags, we trained a Vietnamese POS tagger with the method of Nghiem et al. (2008), using the same training data as the parsers. This POS tagger achieved an accuracy of 94.65% on the test set.

[6]An anchor rule consists of the first and last words of spans, span lengths, span shape descriptions, and the start, stop, and split indexes where the rules are anchored.

Parser	F-score	1-Word Phrase	Unary	VP Attach	NP Attach	PP Attach	Clause Attach	NP Int.	Mod Attach	Diff Label	Co-ord	XoverX Unary	Other
Stanford-Unlex	53.49	1.33	0.86	**2.23**	**1.86**	1.80	1.06	0.68	0.78	0.92	0.06	0.03	4.06
Stanford-RNN	64.16	1.09	0.72	**1.87**	**1.33**	1.35	0.98	0.56	0.96	0.88	0.07	0.02	2.49
Berkeley	71.79	0.91	0.54	**1.60**	**1.10**	0.91	0.87	0.44	0.52	0.75	0.04	0.02	1.99
Epic-CRF	72.03	0.94	0.53	**1.48**	**1.16**	0.87	0.83	0.44	0.49	0.88	0.03	0.02	2.09
Epic-NeuralCRF	71.66	0.90	0.53	**1.53**	**1.27**	0.96	0.84	0.44	0.44	0.79	0.04	0.02	2.15
RNNGs-D	70.86	0.92	0.56	**1.54**	**1.25**	1.00	0.92	0.42	0.46	0.72	0.04	0.03	1.79
RNNGs-G	72.22	0.86	0.48	**1.36**	**1.10**	0.94	0.81	0.38	0.38	0.70	0.03	0.02	1.78

Table 2: Average number of bracket errors per sentence on the development set of Vietnamese Treebank.

sequences based on the full input sentences, which helps it find more useful contextual information.

Our evaluation of representative parsing models on the Vietnamese Treebank provided an overall sense of parsing accuracy in the Vietnamese language. While RNNGs-G achieved the highest accuracy, this was still far less than that reported in English or Chinese. In the following sections, we will explain how we investigated parsing errors to find reasons for the poor accuracy of Vietnamese parsing. This investigation was a valuable step toward improving the quality of Vietnamese parsing.

3 Investigating behaviors of parsers

This investigation was aimed at clarifying two main issues: (1) what are the most frequent errors in Vietnamese parsing?; and (2) which errors can be addressed by different parsing methods? To answer these two questions, we used a tool developed by Kummerfeld et al. (2012)[7] to classify the parsing errors into error types such as VP and NP attachments. We then computed the average number of bracket errors per sentence for each error type. The results on the development set of the Vietnamese treebank are presented in Table 2[8].

By comparing eleven types of errors produced by the parsers (columns from “1-Word Phrase” to “XoverX Unary" in Table 2), we can see that bracket errors caused by VP attachments are the most frequent, while the ones caused by NP and PP attachments are the second most. The table also indicates how the errors can be addressed by different parsing methods. For example, RNNGs-G is the best method that could address most error types in Vietnamese parsing.

In comparison to English parsing errors (Kummerfeld et al., 2012), errors in Vietnamese have the following attributes. (i) Most error types that appeared in English parsing also appeared in Vietnamese parsing. However, the average errors per sentence in Vietnamese parsing were much larger than those in English, except for co-ordination. (ii) PP-attachment was the most problematic constructions in English but it was not in Vietnamese (it was ranked at the third position). (iii) VP attachment was the most frequent error type in Vietnamese parsing while it did not occur in English parsing.

Kummerfeld et al. (2013) also analyzed Chinese parsing errors but we did not directly compare with them because our method of error classification is different from their method. Nevertheless, we drew three main conclusions from our analysis results: (i) the NP internal structure is the most problematic construction in Chinese parsing, while it is rare in Vietnamese parsing. (ii) Although VP, NP, and PP attachments are the top three difficult constructions in Vietnamese parsing, they are not frequent errors in Chinese parsing; no NP attachment errors were even detected in Chinese parsing. (iii) One-word phrases are the most problematic constructions in both languages.

Apart from containing the same complex constructions as English and Chinese such as PP attachments, Vietnamese parsing has its own constructions, e.g., VP attachments, that cannot be solved with the current parsing methods. A new parsing method to address such specific issues in the Vietnamese language is required. By quantifying error types produced by different parsing methods, our investigation provided an orientation for improving the quality of Vietnamese parsing. For example, solving the most frequent errors, such as VP attachment, NP attachment, and PP attachment errors can significantly improve the parsing performance. However, we could not find reasons of error types from this investi-

[7]This tool was designed to classify English parsing errors. We found about 20.1% of errors were identified as the “Other” type by directly applying this tool to Vietnamese parsing output (this ratio is about 11.3% for English). Several errors in the “Other” class could be classified into existing types such as clause and VP attachments.

[8]The types of errors reported in this paper are the same as those in Kummerfeld et al. (2012).

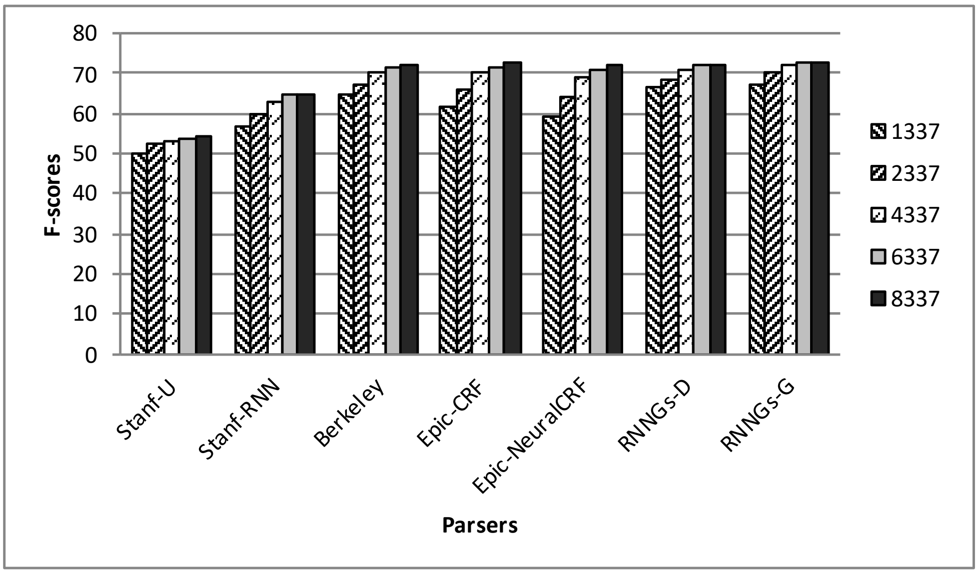


Figure 1: Parsing results from the parsers with different amounts of training data.

gation, which are very important clues to solving errors. We therefore conducted another investigation on three possible sources for the poor performance of Vietnamese parsing in the following sections, viz., the impact of training data size (Section 4), impact of POS tagging errors (Section 5), and impact of ambiguous constructions (Section 6).

4 Impact of training data size

Our training data consisted of 8,337 sentences, which were much fewer than the data used for English and Chinese parsing. To verify whether the small size of training data was a reason for the poor accuracy of the parsers, we evaluated them on different amounts of training data including 1,337, 2,337, 4,337, 6,337, and 8,337 sentences. The parsing results (F_1) on the test set of the Vietnamese Treebank are presented in Figure 1. The figure shows that the parsers' performance was improved when we eventually increased the training data from 1,337 to 6,337. However, the F-score is almost saturated when the number of sentences was increased from 6,337 to 8,337 sentences. These observations indicated that we could not expect a significant improvement in accuracy by simply increasing the amount of data. Consequently, the amount of training data was not the main reason for the poor performance of the parsers.

5 Impact of tagging errors

Figure 2 presents the PARSEVAL evaluations (F-scores) on two different versions of the test set, *PredictedPOS*, i.e., POS tags were automatically produced, and *GoldPOS*, i.e., gold POS tags were used. These results indicate that using gold POS tags could improve the accuracies of the parsers. The F-score of RNNGs-G especially achieved 78.2%, which was improved by 5.26% points in comparison with the use of an automatic POS tagger (*PredictedPOS*). However, this figure does not indicate the error types that could be solved by improving POS tagging.

To find how the gold POS tags affected parsing errors, we compared the error types produced by the best parser, RNNGs-G, on two different versions of the development set: (1) *Gold*: by using gold word segmentation and POS tags and (2) *Pred.*: by only using gold word segmentation. Results in Table 3 reveal that the occurrence of some error types was reduced when gold POS tags were used. However, it should be noted that the reductions were not significant, especially in the cases of VP and NP attachments. Therefore, enhancing POS tagger would somehow improve the performance of Vietnamese parsing but the improvement would not be radical since VP and NP attachments are the most frequent errors.

We used the same method as Kummerfeld et al. (2013) to find how an ambiguous POS pair impacted

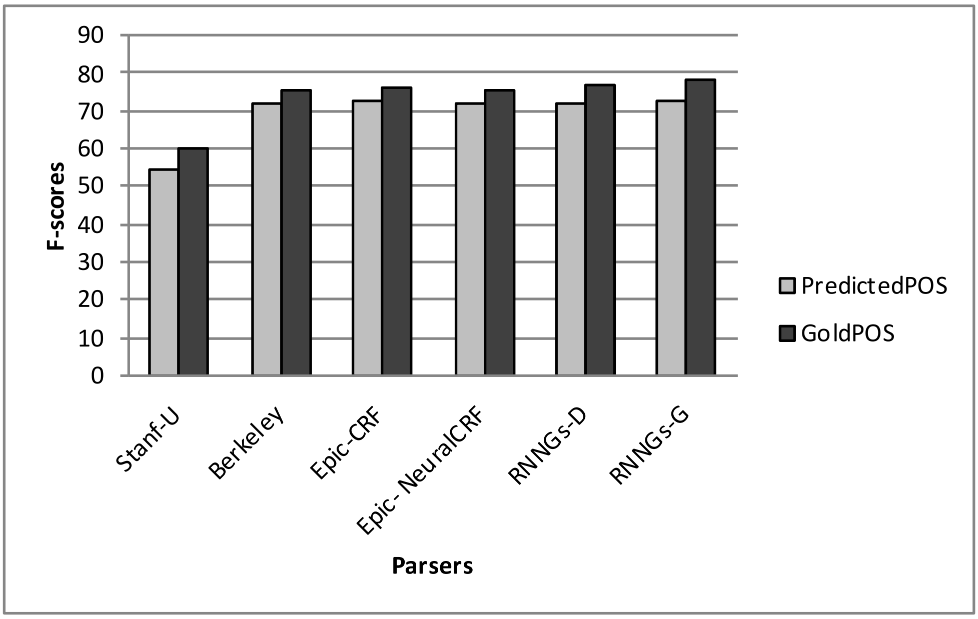


Figure 2: Results from the parsers on two different versions of test set, PredictedPOS and GoldPOS.

parsing errors. We began from *Gold*, and replaced the gold tags with tags predicted by the automatic tagger (semi-gold input). We then parsed the semi-gold input by using RNNGs-G and obtained the results from the PARSEVAL evaluation and automatic analysis. The impacts of the top four confusing POS pairs on overall bracketing errors and the F-scores are presented in Table 4. We can see that all of these four confusing POS pairs were potential contributors to bracketing errors that caused significant decreases in F-scores. However, this table also indicates that the frequency of confusing POS pairs was not directly proportional to the contributions they made to parsing errors. For example, although *Aa/Vv* was the third most confusing POS pair, it contributed most to parsing errors and caused the highest decrease in F-scores (2.58). While occurrences of *Nn/Vv* were highest (106 times), they caused the lowest decrease in F-scores (1.94).

Table 5 summarizes the impact that the top four confusing POS pairs had on each error type. The positive and negative numbers correspond to the number of bracketing errors that were created or reduced in parsing output when we replaced gold POS tags with predicted POS tags. Analysis based on four of the most frequent error types of VP, NP, PP, and clause attachments revealed the following.

VP and NP attachments: As previously explained, improving POS tagger did not assist in preventing VP and NP attachment errors. We can see from Table 5 that bracketing errors caused by VP attachment increased when we replaced the gold tag of *Vv* with predicted tags of *Nn* or *Aa*. Bracketing errors caused by NP attachment also increased when we revised the gold tag of *Aa* with the predicted tag of *Vv*. This indicated that gold POS tags were helpful in these cases. However, using predicted POS tags was better for some constructions, e.g., using the predicted tag of *Vv* instead of the gold *Nn* reduced 65 bracketing errors that were caused by NP attachment errors. These observations have revealed that the gold POS tags are important for solving VP and NP attachment errors. However, it is possible that the current 33 POS tags of the Vietnamese Treebank are not good enough for disambiguating constructions.

PP attachments: All four confusing POS pairs caused these types of parsing errors. However, *Aa/Vv* was the major contributor to parsing errors

Clause attachments: *Aa/Vv* was the major contributor to these types of parsing errors. In addition, ambiguous POS pairs, such as *Nn/Vv* and *Vv/Aa*, also created significant numbers of bracketing errors.

In summary, the quality of Vietnamese parsing could be improved by about 5% points through enhancing the Vietnamese POS tagging. We also found that all frequent ambiguous POS pairs contributed to parsing errors, in which *Aa/Vv* was the major contributor. Our analysis results indicated that improving POS tagging was beneficial for some constructions, such as PP and clause attachments. However,

Error type	Occurrences		Node errors		Node errors per sentence		Gain
	Pred.	Gold	Pred.	Gold	Pred.	Gold	
1-Word Phrase	765	517	857	544	0.86	0.54	0.32
Unary	475	386	475	386	0.48	0.39	0.09
VP Attach	392	373	1362	1368	1.36	1.37	-0.01
PP Attach	415	345	936	837	0.94	0.84	0.10
Diff label	350	157	700	314	0.70	0.31	0.39
NP Attach	333	339	1104	1101	1.10	1.10	0.00
Clause Attach	327	277	814	696	0.81	0.70	0.11
Mod Attach	216	182	381	362	0.38	0.36	0.02
NP Int.	245	211	379	324	0.38	0.32	0.06
Co-ord	34	44	34	44	0.03	0.04	-0.01
XoverX Unary	24	25	24	25	0.02	0.03	0.01
Other	969	729	1779	1435	1.78	1.44	0.34
Sum	4545	3585	8845	7436			

Table 3: Statistics on errors produced by RNNGs-G on two different versions of the development set, *Pred.* and *Gold*. "Gain" represents gain (positive number) and loss (negative number) of errors per sentence when replacing automatically predicted POS tags with gold POS tags (i.e., Errors per sentence (Pred.) - Errors per sentence (Gold)).

Confusing tags	Occurrences	Bracketing errors	F1	Δ F1
Nn/Vv	106	7694	76.26	-1.94
Vv/Nn	87	7764	76.15	-2.05
Aa/Vv	77	7897	75.62	-2.58
Vv/Aa	69	7749	76.07	-2.13
Gold		7436	78.20	

Table 4: The most frequently confusing POS tag pairs in Vietnamese POS tagging. Δ F1 indicates the decreases in F-scores when gold POS tags were replaced with predicted POS tags. Meanings of POS tags are: *Nn*: common nouns, *Vv*: common verbs, and *Aa*: adjectives. For each confusing POS pair *x/y*, *x* is the gold POS, and *y* is the predicted one.

not all gold tags are helpful for solving the parsing errors, which means that the current 33 POS tags of the Vietnamese Treebank are not good enough for disambiguating constructions. We therefore should improve the tag set (e.g., splitting tags) or find more features to address the parsing errors, especially VP and NP attachments—the two most frequent error types.

6 Ambiguous constructions in Vietnamese

This section aims to investigate parsing errors caused by ambiguous constructions in Vietnamese. Our investigation was carried out on the *Gold* data set parsed by the RNNGs-G model trained on 8,337 sentences. By doing that, we could eliminate the impacts of limited training data, word segmentation errors, and confusing POS tags from the parsing output.

We randomly selected 100 sentences from the parsing output for this investigation. We then classified

Error type	Nn/Vv	Vv/Nn	Aa/Vv	Vv/Aa
Single Word Phrase	80	70	66	68
Unary	16	22	52	38
VP Attachment	-26	44	-45	17
NP Attachment	-65	-45	37	-43
PP Attachment	27	51	83	48
Clause Attachment	75	26	103	70
NP Internal Structure	69	21	50	30
Modifier Attachment	11	25	-11	-14
Different label	42	44	32	22
Co-ordination	-21	-14	-14	-21
XoverX Unary	-1	-1	-2	-1
Other	51	85	110	99

Table 5: Gains and Losses in bracket errors when gold POS tags were replaced with predicted ones.

Ambiguous construction	Error type	Frequency
Nx|Vv|Aa|Cs Vv|Aa|Nx VP	VP attachment	19/34
Vv|Nx Vv|Aa|Nx NP	NP attachment	25/39
Nx|Vv Nx|Vv PP	PP attachment	17/29
NP VP NP VP	Clause attachment	10/26
NP|VP NP|VP VP	Clause attachment	6/26
Cs NP VP	Clause attachment	4/26

Table 6: Several frequent ambiguous constructions in Vietnamese parsing.

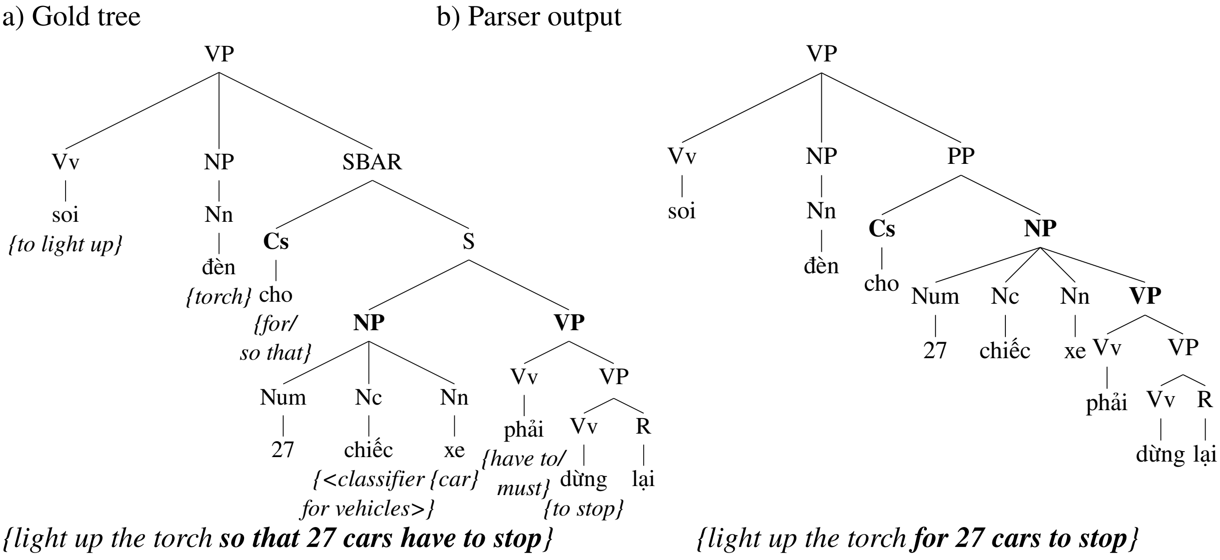


Figure 3: Examples illustrating an VP attachment error that is caused by ambiguous construction *Cs NP VP* in Vietnamese.

parsing errors into error types by applying the automatic analysis tool (Kummerfeld et al., 2012) to the selected sentences. Next, we manually investigated the reasons for each error type. Since individual error types were probably caused by particular constructions, finding reasons for parsing errors according to their error types would provide precise orientations for solving the errors.

Table 6[9] presents several frequent ambiguous constructions[10] in Vietnamese parsing that were found in this manual investigation. The column "Ambiguous construction" in this table presents frequent ambiguous tag sequences in Vietnamese. Error types caused by ambiguous tag sequences are indicated in the column "Error type". For example, tag sequences in the first row, such as *Cs Nx VP*, are candidates for VP attachment errors. For each pair of *x/y* in the column "Frequency", *x* represents the number of times that each ambiguous tag sequence caused errors in 100 investigated sentences; while *y* denotes the frequency of each error type in 100 investigated sentences. For instance, 19/34 in the first row means that we found 34 VP attachment errors in 100 investigated sentences, in which 19 errors were caused by the ambiguous constructions listed in the first column.

These ambiguous constructions are caused by the characteristics of the Vietnamese language, such as lack of inflectional morphemes, polysemous words, word order in which modifying lexical words follow the head word, and word omission.

Figure 3[11] illustrates an ambiguity attributed by most of the above-mentioned linguistic features. In

[9] *Nx* in Table 6 represents a noun that can be *Nn* (a common noun), *Nc* (a classifier noun), or *Ncs* (a special classifier noun). Each component of the tag sequence can be generalized as a phrase, e.g., instead of *Nn*, a noun phrase with head word *Nn* can be placed in the position of *Nn*.

[10] Refer to Appendix D for more details about the ambiguous constructions in Vietnamese parsing.

[11] Underscore "_" is used to link syllables of Vietnamese multi-syllabic words. English translations of Vietnamese words are provided as subscripts. If a Vietnamese word does not have a translatable meaning, the subscript is blank. The translation for the Vietnamese sentence is provided in braces below the original text.

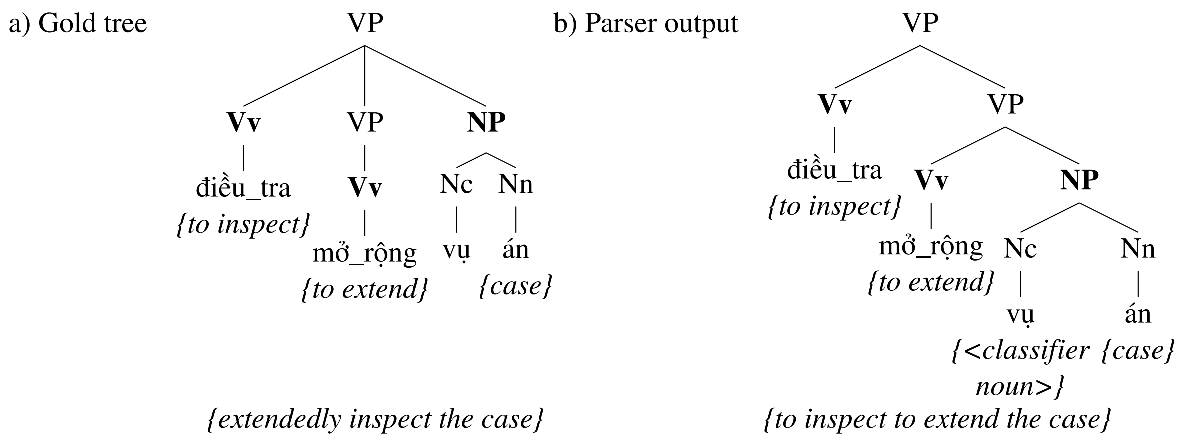


Figure 4: Examples illustrating an NP attachment error that is caused by ambiguous construction *Vv Vv NP* in Vietnamese.

the figure, the bold *NP VP* should be considered as a simple sentence (gold tree—Figure 3a) where *VP* is a predicate, but the parser labelled the whole construction as a noun phrase (*NP*) in which *VP* is an attachment (parser output—Figure 3b). A reason for that is the lack of inflectional morphemes in Vietnamese, in which a verb can (1) modify a noun, a verb, or an adjective, or (2) be the main verb of a sentence, without changing the word form. In addition, modifying verbs, adjectives, etc. follow the head words in Vietnamese noun phrases, which makes the word order in noun phrases identical to that in simple sentences. Polysemous words are also a possible reason for such parsing errors. In the figure, the word *"cho"* has two different interpretations. It means *"so that"* in the case it precedes a clause as shown in the gold tree. When it precedes a noun phrase, it means *"for"* as shown in the parser output. It is difficult for RNNGs-G to parse this case because both constructions *"cho S"* and *"cho NP"* occur in the Vietnamese Treebank.

Figure 4 presents the ambiguous construction of *Vv Vv NP* (row 2 in Table 6), a candidate for the NP attachment error in Vietnamese parsing. The *NP* in this construction can be annotated as the modifier of the first verb (gold tree) or the modifier of the second verb (parser output). This ambiguity is also caused by the common word order in Vietnamese, where modifying lexical words follow the head word. In addition, this also results from the lack of inflectional morphemes. We can see from the figure that this is not an ambiguous construction in English (or Chinese) parsing because cases like *"mở_rộng*$_{to\ extend}$*"* are expressed by adverbs that appear before the head verb and cannot be combined with a noun. However, Vietnamese does not have adverbs; verbs and adjectives are used in such contexts. It is confusing to bracket these constructions because a noun phrase can be the modifier of both previous words.

Statistics on the top four frequent error types, i.e., VP, NP, PP, and clause attachments, revealed that about 63% of attachment errors were caused by ambiguous constructions in Vietnamese. Although RNNGs-G considered the full input sentence in its model, the contextual information was only found based on the coarse non-terminal symbols and surface forms of the words, which were not enough to address ambiguous constructions.

We found in Section 2 that non-neural Berkeley and Epic parsers perform competitively to RNNGs-G. This might suggest that these two parsers have their own advantages, and a possible direction for the future development of Vietnamese parsers is to integrate the idea of these parsers with the modern strong neural parsers. For example, although RNNGs represent each constituent in a continuous space, explicit, symbolic distinction among categories as done in the Berkeley and Epic parsers would still be useful for Vietnamese parsing. In the Vietnamese treebank, subordinate conjunctions that introduce a clause and a noun phrase are both annotated with the same POS tag of *Cs*. This caused ambiguous constructions in which a relative clause and a prepositional phrase have the same tag sequence of *Cs NP VP*. Symbol splitting might help for resolving this ambiguity. For example, we can refine the POS tag for *"cho"* mentioned in Figure 3 as *Cs-SBAR* and *Cs-PP*, depending on whether it precedes a sentence or a noun phrase, with manual refinements as in Stanford and Epic parsers, or automatic refinements as in the Berkeley parser. We therefore expect that one possible approach for future research is a hybrid approach that combines the advantages of different parsers.

7 Conclusion

We have evaluated representative parsing models on the Vietnamese Treebank and found that parsing models based on CRF and neural networks are good for Vietnamese. Contextual information is very beneficial for improving the parsing performance. We have also investigated the errors produced by the parsers. This investigation has revealed the frequent errors in Vietnamese parsing: VP attachment, NP attachment, PP attachment, and clause attachment. Although POS tagging errors have significantly contributed to many parsing errors such as PP attachment and clause attachment errors, they are not an essential reason for the top two error types, VP attachment and NP attachment.

In addition, our investigation on parsing errors has found that ambiguous constructions are the major contributor to the errors in Vietnamese parsing, which are resulted from the characteristics of the Vietnamese language. Our goal in future work is to develop a parser that can overcome the challenges found by the analysis in this paper, and one possible direction is a hybrid approach that combines the advantages of different parsing models investigated in the paper.

References

Anh-Cuong AC. Le, Phuong-Thai Nguyen, Hoai-Thu Vuong, Minh-Thu Pham, and Tu-Bao Ho. 2009. An experimental study on lexicalized statistical parsing for vietnamese. In *Proceedings of Knowledge and Systems Engineering*, pages 162–167. IEEE.

Daniel M Bikel. 2004. *On the parameter space of generative lexicalized statistical parsing models*. Ph.D. thesis, Citeseer.

Maximin Coavoux and Benoit Crabbé. 2017. Incremental discontinuous phrase structure parsing with the gap transition. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, volume 1, pages 1259–1270.

Greg Durrett and Dan Klein. 2015. Neural crf parsing. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, page 302–312. Association for Computational Linguistics.

Chris Dyer, Adhiguna Kuncoro, Miguel Ballesteros, and Noah A Smith. 2016. Recurrent neural network grammars. In *Proceedings of NAACL-HLT 2016*, page 199–209. Association for Computational Linguistics.

Jenny Rose Finkel, Alex Kleeman, and Christopher D Manning. 2008. Efficient, feature-based, conditional random field parsing. In *ACL*, volume 46, pages 959–967.

David Hall, Greg Durrett, and Dan Klein. 2014. Less grammar, more features. In *Proceedings of the 52rd Annual Meeting of the Association for Computational Linguistics*, pages 228–237.

Tadayoshi Hara, Yusuke Miyao, and Jun'ichi Tsujii. 2009. Descriptive and empirical approaches to capturing underlying dependencies among parsing errors. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3-Volume 3*, pages 1162–1171. Association for Computational Linguistics.

Dan Klein and Christopher D Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 423–430. Association for Computational Linguistics.

Jonathan K Kummerfeld, David Hall, James R Curran, and Dan Klein. 2012. Parser showdown at the wall street corral: An empirical investigation of error types in parser output. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1048–1059. Association for Computational Linguistics.

Jonathan K Kummerfeld, Daniel Tse, James R Curran, and Dan Klein. 2013. An empirical examination of challenges in chinese parsing. In *ACL (2)*, pages 98–103.

Phuong Le-Hong, Thi-Minh-Huyen Nguyen, and Azim Roussanaly. 2012. Vietnamese parsing with an automatically extracted tree-adjoining grammar. In *Computing and Communication Technologies, Research, Innovation, and Vision for the Future (RIVF), 2012 IEEE RIVF International Conference on*, pages 1–6. IEEE.

Roger Levy and Christopher Manning. 2003. Is it harder to parse chinese, or the chinese treebank? In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 439–446. Association for Computational Linguistics.

Takuya Matsuzaki, Yusuke Miyao, and Jun'ichi Tsujii. 2005. Probabilistic cfg with latent annotations. In *Proceedings of the 43rd annual meeting on Association for Computational Linguistics*, pages 75–82. Association for Computational Linguistics.

Minh Nghiem, Dien Dinh, and Mai Nguyen. 2008. Improving vietnamese pos tagging by integrating a rich feature set and support vector machines. In *Proceedings of Research, Innovation and Vision for the Future in Computing and Communication Technologies (RIVF)*, pages 128–133. IEEE.

Phuong-Thai Nguyen, Xuan-Luong Vu, Thi-Minh-Huyen Nguyen, Van-Hiep Nguyen, and Hong-Phuong Le. 2009. Building a large syntactically-annotated corpus of vietnamese. In *Proceedings of the Third Linguistic Annotation Workshop*, pages 182–185. Association for Computational Linguistics.

Quy T. Nguyen, Yusuke Miyao, Ha T.T. Le, and Ngan L.T. Nguyen. 2016. Challenges and solutions for consistent annotation of vietnamese treebank. In *Proceedings of the Language Resources and Evaluation Conference*.

Slav Petrov and Dan Klein. 2007. Improved inference for unlexicalized parsing. In *HLT-NAACL*, volume 7, pages 404–411. Association for Computational Linguistics.

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

Richard Socher, John Bauer, Christopher D Manning, and Andrew Y Ng. 2013. Parsing with compositional vector grammars. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 455–465. Association for Computational Linguistics.

Fei Xia, Martha Palmer, Nianwen Xue, Mary Ellen Okurowski, John Kovarik, Fu-Dong Chiou, Shizhe Huang, Tony Kroch, and Mitchell P Marcus. 2000. Developing guidelines and ensuring consistency for chinese text annotation. In *Proceedings of the Second International Conference on Language Resources and Evaluation*.

Muhua Zhu, Yue Zhang, Wenliang Chen, Min Zhang, and Jingbo Zhu. 2013. Fast and accurate shift-reduce constituent parsing. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 434–443.

Appendix A The POS tags designed for the Vietnamese Treebank

No.	POS tag	Meaning of tag	No.	POS tag	Meaning of tag
1	SV	Sino-Vietnamese syllable	17	NA	Noun-adjective
			18	Vcp	Comparative verb
2	Nc	Classifier noun	19	Vv	Other verb
3	Ncs	Special classifier noun	20	An	Ordinal number
4	Nu	Unit noun	21	Aa	Other adjective
5	Nun	Special unit noun	22	Pd	Demonstrative pronoun
6	Nw	Quantifier indicating the whole	23	Pp	Other pronoun
			24	R	Adjunct
7	Num	Number	25	Cs	Preposition or conjunction introducing a clause
8	Nq	Other quantifier			
9	Nr	Proper noun	26	Cp	Other conjunction
10	Nt	Noun of time	27	ON	Onomatopoeia
11	Nn	Other noun	28	ID	Idioms
12	Ve	Exitting verb	29	E	Exclamation word
13	Vc	Copula "là" verb	30	M	Modifier word
14	D	Directional verb	31	FW	Foreign word
15	VA	Verb-adjective	32	X	Unidentified word
16	VN	Verb-noun	33	PU	Punctuation

Appendix B The constituency tags designed for the Vietnamese Treebank

No.	Tag	Explanation
1	NP	Noun phrase
2	QP	Quantitative phrase
3	VP	Verb phrase
4	ADJP	Adjective phrase
5	PP	Prepositional phrase
6	RP	Adjunct phrase
7	CONJP	Conjunction phrase
8	UCP	Unlike coordinated phrase
9	QNP	Questioning noun phrase
10	QRP	Questioning adjunct phrase
11	QPP	Questioning prepositional phrase
12	QADJP	Questioning adjective phrase
13	MDP	Modal phrase
14	S	Simple/compound declarative sentence
15	SQ	Question
16	SPL	Special sentence
17	SBAR	Subordinate clause
18	XP	Unknown phrase

Appendix C Examples illustrating Vietnamese characteristics

Figure 5 represents an example illustrating two characteristics of Vietnamese, viz., lack of inflectional morphemes and the word orders. We can see from English translations in this figure that the noun *establishment* (Figure 5a) and the past tense verb *established* (Figure 5b) can be distinguished on the basis of inflectional morphemes *"-ment"* and *"-ed"*. However, Vietnamese does not have such clues because Vietnamese is a non-inflected language. This leads to the situation that the verbs *thiết_lập*$_{to\ establish}$ have the same surface forms in both syntactic roles, i.e., the modifier of the noun *thủ_tục*$_{procedure}$

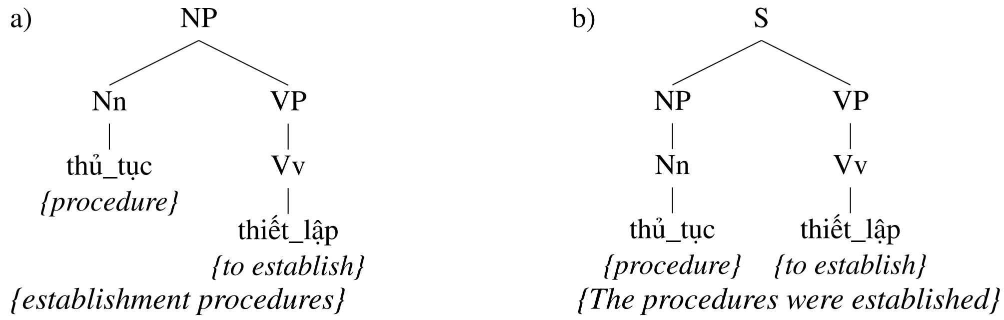


Figure 5: Examples illustrating Vietnamese characteristics.

a) Gold tree

b) Parser output

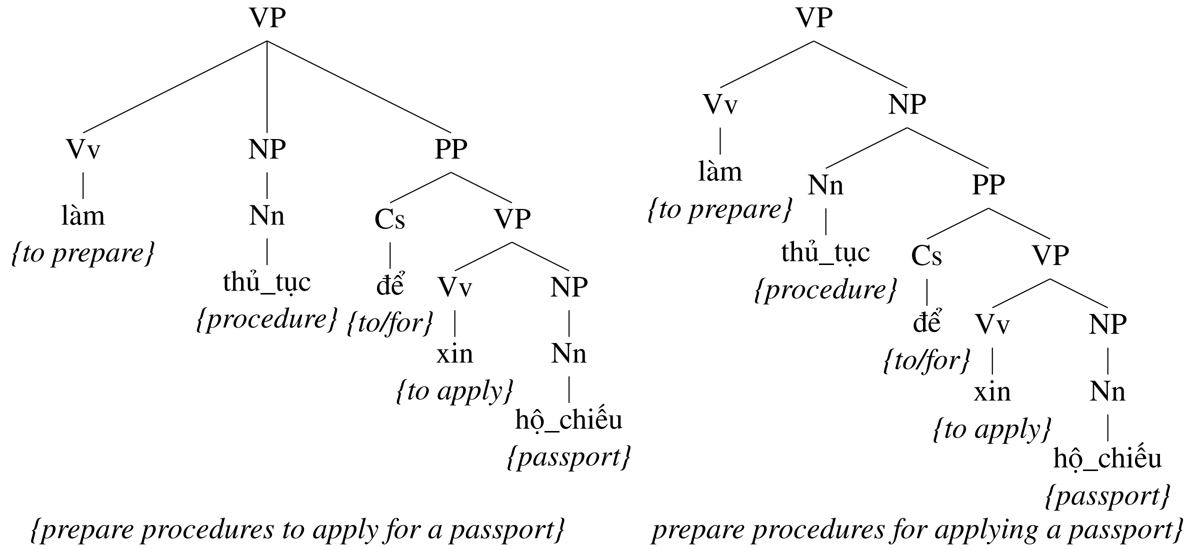


Figure 6: Examples illustrating the PP attachment error in Vietnamese parsing.

(Figure 5a) and the predicate of the simple sentence (Figure 5b). In addition, the modifying verb follows the head noun in a Vietnamese noun phrase (as shown in Figure 5a). This word order is the same as that in a simple sentence (Figure 5b). Parsing noun phrases and simple sentences is ambiguous because they have the same construction of *Nn Vv*.

Appendix D Examples illustrating several error types in Vietnamese parsing

As mentioned in the main content of the paper, we applied the error detection tool proposed by Kummerfeld et al. (2012) to Vietnamese. Therefore, our error types are the same as those in Kummerfeld et al. (2012). Follows are more details about the top four frequent error types in Vietnamese parsing.

D.1 VP attachment error

Constructions in which *VPs* are moved, viz., the incorrect bracket over an *VP* or incorrect attachment in noun phrases, verb phrases, adjective phrases, and prepositional phrases (an example was given in Section 6).

D.2 NP attachment error

Constructions in which *NPs* are moved (an example was given in Section 6).

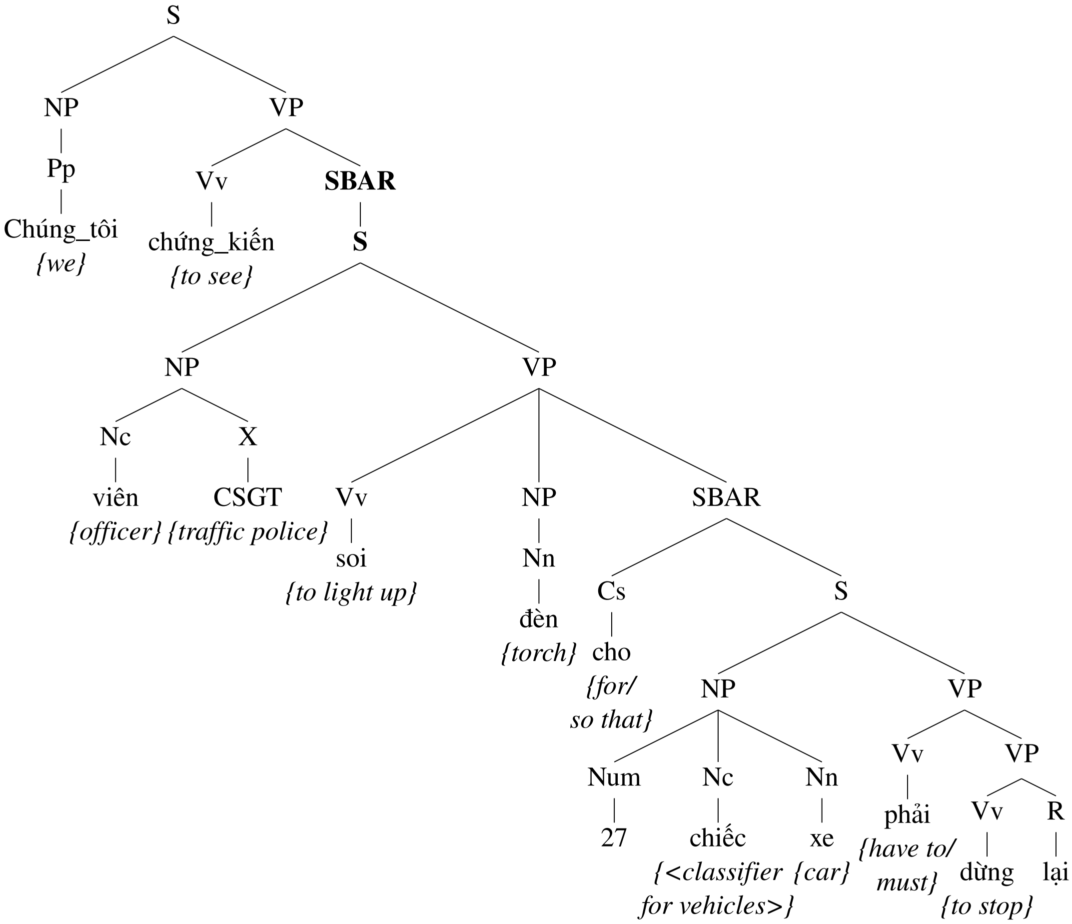
D.3 PP attachment error

Constructions in which *PPs* are moved. For example, Figure 6 represents an PP attachment error caused by the ambiguous construction *Vv Nn PP*. The *PP* in this example should be annotated as a modifier of the previous verb *làm*$_{to\ prepare}$ (gold tree), rather than attached to the previous noun *thủ_tục*$_{procedure}$ (parser output).

D.4 Clause attachment error

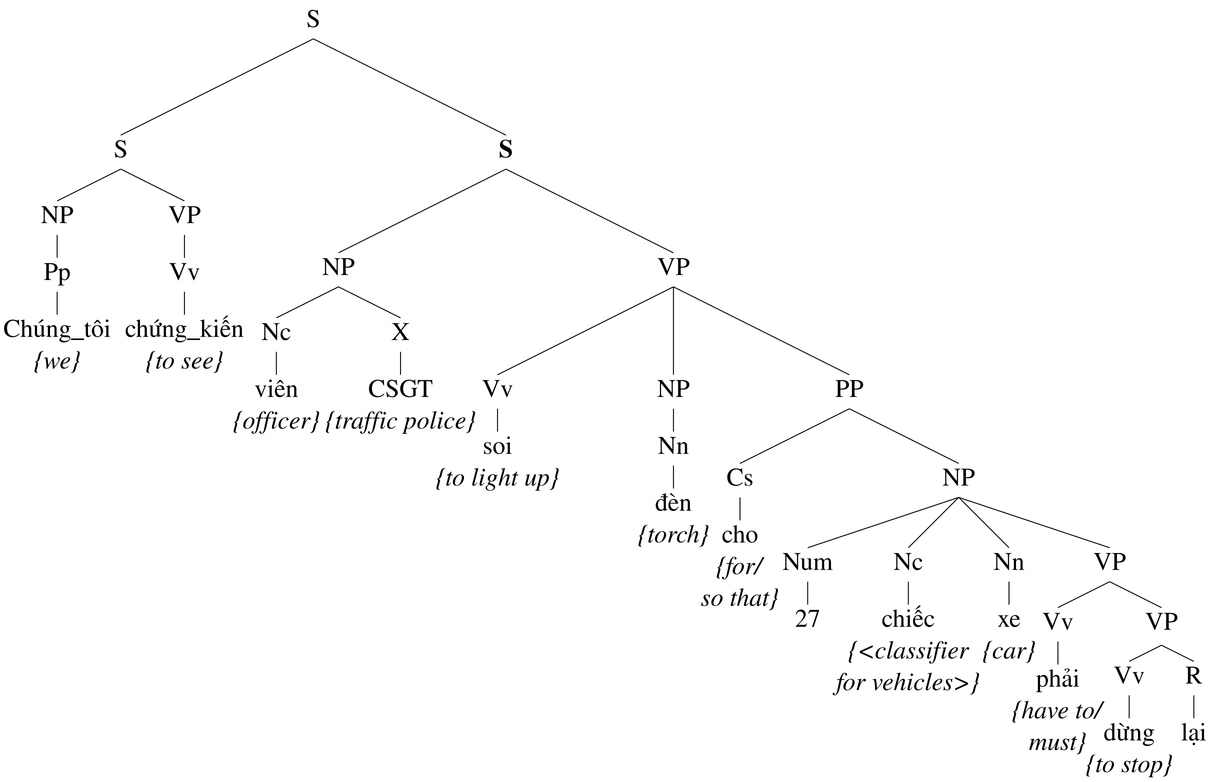
Constructions in which *Ss* or *SBARs* are moved. Figure 7 represents an example of clause attachment error caused by the ambiguous construction of *NP VP NP VP*. We can see that the later *NP VP* should be considered as an *SBAR* modifying the previous verb *chứng_kiến*$_{to\ see}$ (gold tree). However, it was attached as the second clause of a compound sentence (parser output).

a) Gold tree



{We saw that the traffic police officer lighted up the torch so that 27 cars have to stop}

b) Parser output



{We saw the traffic police officer lighted up the torch for 27 cars to stop}

Figure 7: Examples illustrating the clause attachment error in Vietnamese parsing.