ICIC Express Letters
Part B: Applications
Volume **13**, Number **3**, March **2022**

ICIC International ©2022 ISSN 2185-2766
pp. **249–259**

APPLYING CROSS-VIEW TRAINING FOR DEPENDENCY PARSING IN VIETNAMESE

Duc Huu Trinh[1,2], Trinh Le-Phuong Ngo[1,2], Long Hong Buu Nguyen[1,2,*] and Dien Dinh[1,2]

[1]Faculty of Information Technology
University of Science, Ho Chi Minh City
227 Nguyen Van Cu Street, Ward 4, District 5, Ho Chi Minh City, Vietnam
{ thduc17; nlptrinh }@apcs.fitus.edu.vn; ddien@fit.hcmus.edu.vn
*Corresponding author: nhblong@fit.hcmus.edu.vn

[2]Vietnam National University, Ho Chi Minh City
Linh Trung Ward, Thu Duc District, Ho Chi Minh City, Vietnam

Received July 2021; accepted October 2021

Abstract. *Dependency parsing aims to identify syntactic relations or dependencies between word pairs in the sentence. Recent research has shown that contextual model like BERT implicitly captures linguistics information, e.g., syntax, and semantic at different hidden layers. In addition, as dependency parsing can be formulated as a classification problem, supervised deep learning models have recently outperformed other methods. However, these supervised models demand a large labeled dataset which is the major drawback of low-resource languages like Vietnamese. To address both issues, we introduce a model that includes two main contributions. First, we integrate PhoBERT at a selective hidden layer to capture syntactic features of the contextual representations. Second, due to the scarcity of labeled data, for which annotation is costly, we examine and adapt cross-view training (CVT) as a semi-supervised learning strategy, allowing us to enhance model representations using unlabeled data. Experimental results on the Vietnamese treebank for dependency parsing and the set of raw sentences taken from Vietnamese News, measured on unlabeled attachment score (UAS) and labeled attachment score (LAS), respectively, show that our model achieved 85.37% and 78.40%, which is 0.42+% and 0.23+% higher than the current SOTA.*

Keywords: NLP, Dependency parsing, Pre-trained language model, PhoBERT, Cross-view training, Semi-supervised, Gold data shortage

1. **Introduction.** Dependency parsing is one of the fundamental problems in natural language processing (NLP) and has proven to boost the performance of many applications, such as information extraction [24], and machine translation [25], due to its ability to capture the syntactic structure of a sentence. The task is described as follows: Given an input sentence, the output is to generate a set of binary grammatical relations that hold between two words, which can be interpreted as directed labeled arcs from the head words to modifier words. These relations can finally draw a tree (usually called dependency tree).

McDonald and Nivre [1] categorized two major approaches of dependency par-sers: graph-based approaches with MSTParser and transition-based ones with MaltParser. For many popular languages such as English or Chinese, recent works using neural network graph-based dependency parser achieve state-of-the-art (SOTA) results, which were pioneered by Kiperwasser and Goldberg [2]. Their parser is the first applied Bi-LSTMs to obtain word vector representation, and then fed them into a classifier to decide their arcs and labels. Later then, Dozat and Manning [3] improved their performance by using a

DOI: 10.24507/icicelb.13.03.249

novel biaffine classifier, which separates the vector representation of a word by two different roles: head and dependent. In the last 2 years, 2019-2020, dependency parsing has witnessed many considerable enhancements [21, 22] by applying attention network instead of LSTM network. However, the larger and deeper the model is, the more data is required. As a result, these methods are not effectively applicable for low-language resource languages, especially Vietnamese.

Despite its importance in NLP, there are only a handful of works on dependency parsing for Vietnamese. Some initial works on dependency parsing have originated from the constituent treebank VietTreebank (VTB) [4]. Phuong et al. [5] trained the VTB data on lexicalized tree-adjoining grammars (LTAG) parser, which returned derivation trees that can extract dependency relations, and achieved 73.21% of UAS using automatically predicted part-of-speech (POS) tags. Thi et al. [6] demonstrated a conversion method from constituency to dependency treebank, and obtained 73.03% UAS and 66.35% LAS on MaltParser using gold standard POS tags. Nguyen et al. [7] provided the first public Vietnamese dependency treebank (VnDT) having about 10,200 sentences, and gained 79.08% in UAS and 71.66% in LAS on MSTParser using gold standard POS tags.

In recent years, pre-trained language deep learning models perform distinctive results as they produce contextualized word embedding. PhoBERT – the first large-scale pre-trained language models for Vietnamese [8] was successfully applied for the joint multi-task learning model PhoNLP [9], bringing PhoNLP to become the current state-of-the-art results on dependency parsing tasks with the UAS and LAS score as 84.95% and 78.17%, respectively.

However, there are still a lot of challenges as well as potential directions for further research in Vietnamese dependency parsing. One of them is the problem of low resources, whose size of treebank is small (public treebank VnDT has 10,200 sentences), making it an issue for improving the parsing performance by using modern deep networks. Some other points are error propagation from POS tag, which is usually used as embedding for dependency parser, or the unification in annotation between existing small treebank.

All of those reasons above have inspired us, so in this paper, we will focus on solving the **limited labeled dataset problem** by applying **cross-view training** – a semi-supervised technique [10]. Besides, we try to analyze the **error sources** of our system and investigate the effectiveness of each **PhoBERT layer** in dependency parsing, while other previous researchers simply used the final layer. Therefore, our models produce new SOTA performance, with 85.37% UAS and 78.40% LAS.

The remainder of the paper is organized as follows. We present the proposed structure of our dependency parser (Section 2), then report the results (Section 3), provide deeper analyses (Section 4), and finally conclude and propose possible orientations for future work (Section 5).

2. Method.

Our dependency parsing model is based on the deep biaffine parser from Qi et al. [15], shown in Figure 1. It is a graph-based method, treating each word as a graph node, and therefore the task's goal becomes to classify head and label for each node. Besides the normally supervised parser, we also introduce how we use the cross-view training technique in Section 2.3.

2.1. Token representation.

In our model, each word in the input data is composed of characters, words, and part-of-speech (POS) tagging representation. We use CharCNN [11] to learn character-level representation of word. For POS tagging, we derived the tool from CLC Lab [12], which assigned POS tag to the input word. For word-level representation, we employ pre-trained model PhoBERT [8] to generate contextualized word embedding. We concatenate the three representations above to obtain a vector representing token: $x_i = [x_{char}; x_{word}; x_{POS}]$.

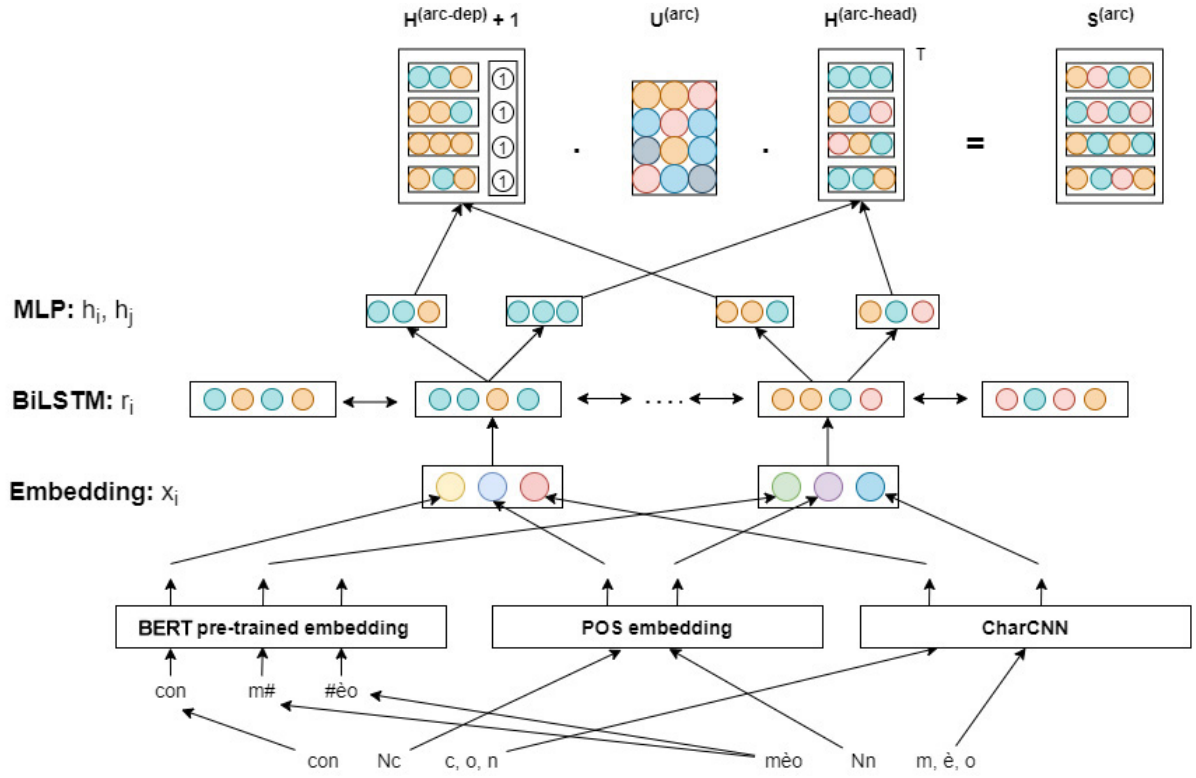


FIGURE 1. Our parser's architecture, including the embedding layer, BiLSTM layer, and the biaffine attention classifier applied to the sentence "Con mèo"

BERT (bidirectional encoder representations from transformers) [13], uses Transformer encoder, which is an attention mechanism that reads the entire sequence of words at once. This characteristic allows the model to learn the context of a word based on the word surroundings (left and right of the word). PhoBERT, the first public large-scale language model [8], using BERT architecture to retrain on Vietnamese dataset, has achieved state-of-the-art results on fundamental tasks like POS tagging, dependency parsing, NER, and NLI.

Recent works [14, 23] have shown that BERT encodes the hierarchical structure of the language in different hidden layers. In particular, the lower layer contains surface information about the word, while the middle layer encodes syntactic features and semantic features at the top. From that motivation and through our experiment, we select a middle layer of PhoBERT as the output layer for our word embedding.

2.2. **Biaffine attention classifier.** After highway BiLSTM [16], each word representation is fed into distinct multi-layer perceptrons (MLP) to be split into head $h_i^{arc\text{-}head}$ or dependent $h_i^{arc\text{-}dep}$ roles. Then the parser applies a bi-linear classifier, with weight parameters $U^{(1)}$ and $u^{(2)}$, on the head and dependent representation to calculate the probability score for each candidate edge (s_i^{arc}).

$$h_i^{arc\text{-}head} = MLP^{arc\text{-}head}(r_i) \tag{1}$$

$$h_i^{arc\text{-}dep} = MLP^{arc\text{-}dep}(r_i) \tag{2}$$

$$s_i^{arc} = H^{arc\text{-}head} U^{(1)} h_i^{arc\text{-}dep} + H^{arc\text{-}head} u^{(2)} \tag{3}$$

The authors also employ another MLPs and classifier for relation type. Moreover, Qi et al. [15] proposed to integrate the additional linear order and distance score explicitly to the distribution of edge score. Finally, Chu-Liu/Edmonds' algorithm [17, 18] was used

to find the maximum spanning tree of the graph, but when training, the cross-entropy loss is computed without forming a tree.

2.3. **Cross-view training.** About semi-supervised algorithms, we follow the method proposed by Clark et al. [10] with a few augmentations in the teacher's input, shown in Figure 2. Besides the primary module (so-called teacher), 5 additional auxiliary parsers (so-called student) are added to our models. Four of them have restricted input views, only seeing the information provided by one direction of the middle layer in the BiLSTM stack, while the final has the same input as the teacher. Particularly, the head and dependent input for each module are

- student 1: forward_layer(middle), forward_layer(middle)
- student 2: forward_layer(middle), backward_layer(middle)
- student 3: backward_layer(middle), forward_layer(middle)
- student 4: backward_layer(middle), backward_layer(middle)
- student 5, teacher: full_layer(final), full_layer(final)

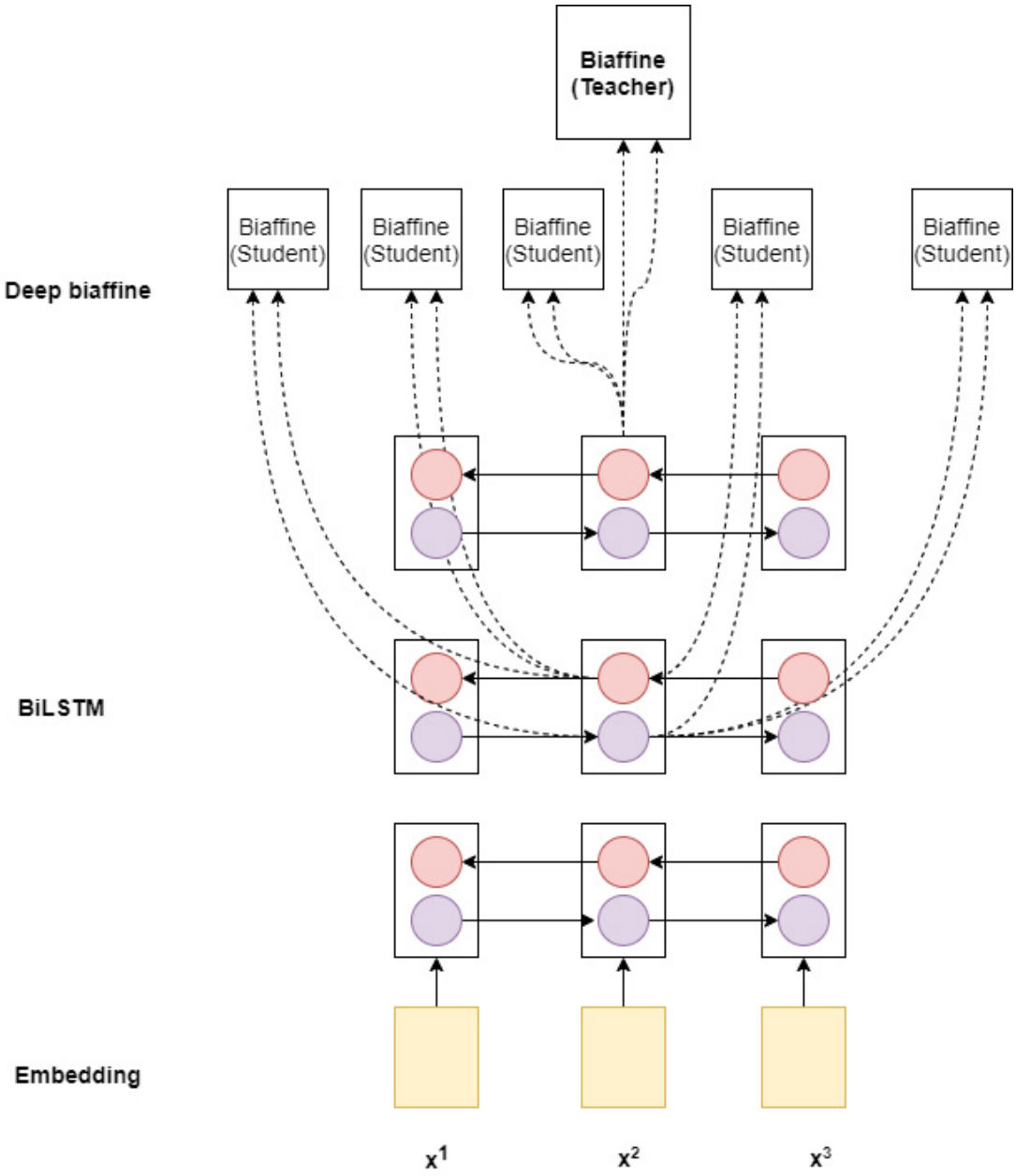


FIGURE 2. The simple visualization of the cross-view training technique

When training, the primary parser learns from the labeled data, and then the auxiliary parser studies the unlabeled source with guidelines predicted by the teacher. The process continuously takes turns with fixed batch between the teacher and students, therefore improving both at the same. Because of sharing the same encoder, BiLSTM layers, and utilizing the enormous raw text, the cross-view training technique is expected to make the encoder more general and the prediction more consistent.

3. **Experiment.**

3.1. **Dataset.** We used the VnDT – a Vietnamese dependency treebank v1.1 to experiment with different components on the model. The treebank was automatically converted from 10,200 sentences (220,000 words) in the constituent treebank VietTreebank [6]. In addition, for cross-view training experiments, we used 300,000 sentences of raw text unlabeled data, taken from Vietnamese Wikipedia & News.

3.2. **Setup.** In our experiment, we used PyTorch to implement our model, and each report below is a mean score over 5 runs. We use an SGD optimizer with momentum followed by the cross-view scheme [10] and train for 40,000 batches with 32 sentences each batch.

3.2.1. *Embeddings.* As we discussed above, our input embedding comprises POS tag embedding, word embedding, and character-level embedding. The dimension size of POS embedding is 50, while our small CharCNN has 3 filter widths, so its total size is 150. About PhoBERT, we perform evaluation cumulatively on the first k layers, for each k ranges from 1 to 12 layers of PhoBERT_base with 768 dimensions, to find the best layer for extracting word representation. Table 1 shows the UAS, LAS score on the baseline model with only PhoBERT taken at different layers as embedding.

Table 1. PhoBERT layer result

Layer No.	UAS	LAS
5	83.94	76.79
6	84.17	76.82
7	84.36	77.01
8	84.57	77.35
9	**84.75**	**77.51**
10	84.5	77.17
11	84.03	76.9
12	84.06	76.83

From Table 1, we can see the language model has increased the baseline by nearly 0.7%, and the output of middle layers, especially layer 9, performs the best. This result shows the fact that middle layers encode best on syntactic information, which is needed for a dependency parsing system. We also try using PhoBERT in our model in 3 different ways: sum, average, and first subwords. Table 2 illustrates that using as the first subword is the best, although their gaps are not significant.

Table 2. PhoBERT using

Method	UAS	LAS
Sum	84.82	77.8
Average	84.68	77.67
First subword	**84.83**	**77.82**

3.2.2. *Parser and cross-view training.* We employ a re-implementation of the variant of deep biaffine parser from Qi et al. [15] with default optimal hyperparameters. However, when experimenting, we found that the distance scorer makes our model easily overfit. Therefore, we just use the additional linear scorer.

- dropout: 0.33
- student dropout: 0.5
- BiLSTM: 3 layers, 400 dimensions
- H_biaffine: 400 dimensions
- base learning rate: 0.5 for teacher, 0.2 for student model

3.2.3. *Result.* Table 3 presents the outcome of 5 different parsers including

- Gold POS: no cross-view training, gold POS
- Non CV: no cross-view training, predicted POS
- CV: cross-view training, predicted POS
- PhoNLP
- PhoNLP single task

TABLE 3. Performance scores (in UAS, LAS) on the test sets of the 5 models

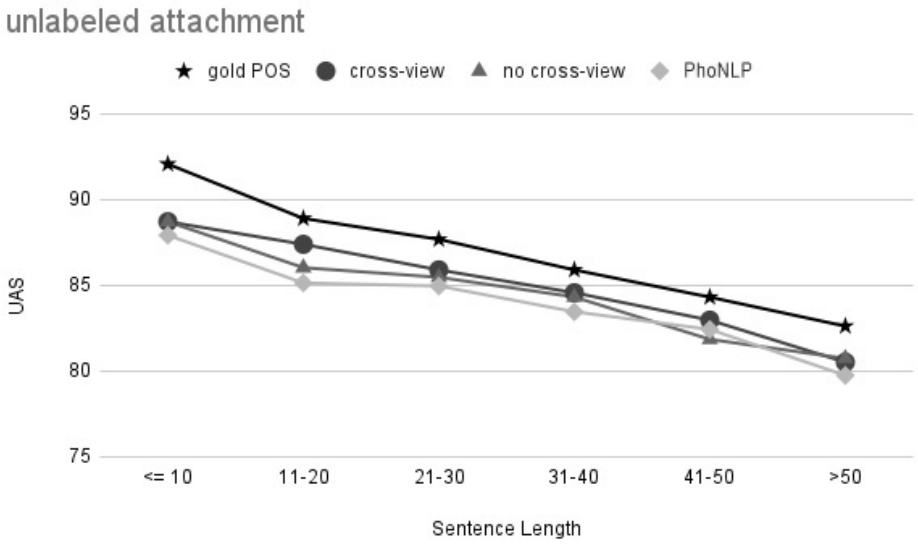
Model	UAS	LAS
No cross-view, gold POS (ours)	**87.19**	**81.81**
No cross-view (ours)	84.83	77.82
Cross-view (ours)	**85.37**	**78.40**
PhoNLP [5]	84.95	78.17
Single task [5]	84.78	77.89

As we can see, the cross-view model's performance is higher than that figures of PhoNLP multi-task model by 0.42 of UAS and 0.23 of LAS. Moreover, when compared with the best single task, it is 0.59 of UAS and 0.51 of LAS. However, all the predicted POS models are still far behind the gold POS model, leading us to many further discussions in the next section.

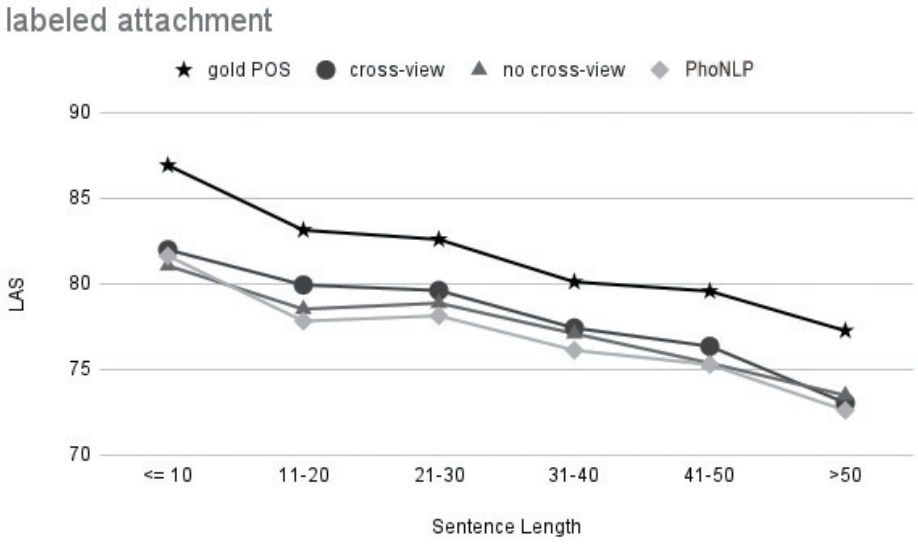
4. **Error Analysis.** This analysis focuses on the four above parsers. The results were measured on the test set of VnDT, reported in LAS, UAS with sentence factor and precision, recall, F1 score following the labeled scheme [19] with linguistic and structural properties. Overall, the gold POS model is the best in most aspects, followed by the cross-view model and finally PhoNLP.

4.1. **Sentence length.** Figures 3(a) and 3(b) illustrate the UAS and LAS of four models relative to 6 groups of sentence length with a bucket of 10 for each group. Understandably, shorter sentences are better in all aspects, but when adding the corrected POS tag, interestingly, it most benefits the shortest and the longest sentence groups, with above 10% in LAS, while the others are 5%.

4.2. **Dependency distance.** Figures 4(a) and 4(b) illustrate F1 scores relative to distance from the dependent word and its head. Similar to a previous study [8], we can see that the left dependencies are predicted better than the right dependencies. Moreover, its gap is even larger in the longer arc or using the corrected POS tag. Another interesting point is that the score of our normal supervised model is as high as our cross-view model in short arcs (absolute value smaller than 2), but it falls back to the same level of PhoNLP in longer distances.



(a)



(b)

FIGURE 3. UAS, LAS by sentence length

4.3. **Part-of-speech tagging.** Table 4 shows the labeled dependency accuracy of 4 parsers for different dependent POS tags along with their percentage. We observe that all parsers achieve high performances for quantity and determiner ($> 92\%$) and low accuracy for preposition, conjunction, abbreviation, and undefined POS tags ($< 70\%$). For POS tags occupying high percentage ($> 1.8\%$), the CV model performs better than the Non-CV model and PhoNLP model but the situation is reversed for the minority POS tags.

4.4. **Dependency type.** Table 5 shows F1 scores, average length, and percentages for the most frequent dependency types, which have popularity higher than 2%.

The data also witness a significant difference between the gold POS and three other models using predicted POS, especially with the relations *vmod*, *amod*, and *adv* (about 10%). That means errors from predicted POS affect a lot to the performance of the downstream dependency parser, while verb and adjectives are two POS tags that suffer

(a)

(b)

FIGURE 4. F1 scores by dependency distance for unlabeled attachment and labeled attachment

the most from this issue. Besides, there are also relations such as *dep* and *tmp*, where the gold model has a huge distance to the PhoNLP model, but its gap to our proposed model is just a half.

4.5. **Size of label and unlabeled dataset.** Table 6 presents the effectiveness of the cross-view training technique relative to its unlabeled dataset size. While both UAS and LAS consistently climb up, following the increase of unlabeled dataset size, dev UAS and LAS fluctuate, but all of them reduce in comparison with not using the cross-view technique. These figures show that a semi-supervised strategy helps solve the overfitting problem.

We also conduct an experiment to present the necessity of the labeled dataset. As we can see in Table 7, when decreasing the size of the labeled train dataset by 25% (2,245

Table 4. Accuracy of models on different dependent POS

POS	Percent	Gold	CV	Non CV	PhoNLP
Noun	30.78	81.99	79.92	79.32	79.48
Verb	18.96	80.71	76.59	75.76	74.13
Punctuation	13.04	80.51	79.87	78.64	76.13
Adjunct	7.25	94.95	86.05	84.61	84.49
Adjective	6.14	79.05	74.43	74.15	73.72
Preposition	6.02	66.55	64.01	63.22	64.59
Pronoun	3.71	87.91	84.98	84.04	84.15
Conjunction	3.62	68.8	64.1	61.33	61.81
Quantity	3.31	94.08	93.82	94.08	92.37
Determiner	1.82	99.04	96.88	97.12	96.4
Particle	0.55	85.71	66.67	70.63	68.25
Un-definition/Other	0.24	48.15	25.93	35.19	37.04
Affix	0.06	92.86	64.29	78.57	64.29
Exclamation	0.04	77.78	66.67	88.89	88.89
Abbreviation	0.01	66.67	66.67	33.33	33.33

Table 5. F1 for different dependency types

Type	Percent	Length	Gold Pos	CV	No CV	PhoNLP
nmod	21.75	1.84	85.58	84.83	84.33	83.57
punct	13.65	8.88	80.49	79.87	78.62	77.61
vmod	12.35	2.56	**73.25**	65.97	65.67	64.32
sub	6.93	3.48	85.54	83.02	81.68	80.71
det	6.08	1.22	96.01	94.88	94.76	94.06
dob	6.02	1.62	74.97	73.51	72.88	74.55
adv	5.89	1.50	**95.28**	85.27	84.94	84.88
pob	5.49	1.31	95.96	91.30	91.62	91.40
root	4.65	5.95	91.57	89.41	88.33	87.65
dep	3.26	7.23	**66.57**	61.03	59.94	56.15
amod	2.67	1.47	**83.02**	73.13	71.70	71.70
tmp	2.14	5.96	**72.26**	68.85	66.20	63.65
loc	2.07	2.70	65.52	63.45	62.46	63.39

Table 6. Unlabeled dataset size

% of unlabeled dataset	UAS	dev UAS	LAS	dev LAS
0%	84.83	**85.70**	**77.82**	78.37
5%	84.95	85.39	77.90	78.07
10%	85.20	85.58	78.12	78.33
100%	**85.37**	85.50	**78.40**	78.23

Table 7. Labeled dataset size

% of labeled dataset	UAS	LAS
100%	84.83	77.82
100%, cross-view	**85.37**	**78.40**
75%	84.36	77.22
75%, cross-view	84.72	77.54

sentences), the performance goes down about 0.5% in all aspects. It also means this task still has a high potential to increase accuracy by adding more labeled data.

5. **Conclusion.** In this paper, we present the effectiveness of the cross-view training technique as well as the appropriate PhoBERT layer on the performance of Vietnamese dependency parsing and analyzing the error sources of this problem. In particular, we achieve new state-of-the-art results on the Vietnamese dependency treebank with 85.37% UAS and 78.40% LAS. According to the analysis outcome, we propose some possible directions to enhance this task, such as reducing the overfitting, enlarging the labeled dataset, minimizing the error propagation from POS, and integrating NER information. Finally, we hope our study will encourage further research and application to utilize the huge available unlabeled data.

Acknowledgment. This research is supported by research funding from the Advanced Program in Computer Science, University of Science, Vietnam National University – Ho Chi Minh City.

REFERENCES

[1] R. McDonald and J. Nivre, Characterizing the errors of data-driven dependency parsing models, *EMNLP-CoNLL*, pp.122-131, 2007.
[2] E. Kiperwasser and Y. Goldberg, Easy-first dependency parsing with hierarchical tree LSTMs, *TACL*, vol.4, pp.445-461, 2016.
[3] T. Dozat and C. D. Manning, Deep biaffine attention for neural dependency parsing, *ICLR2017*, 2017.
[4] P. T. Nguyen, X. L. Vu, T. M. H. Nguyen, V. H. Nguyen and H. P. Le, Building a large syntactically-annotated corpus of Vietnamese, *Proc. of the LAW III*, pp.182-185, 2009.
[5] L.-H. Phuong, H. Nguyen and A. Roussanaly, Vietnamese parsing with an automatically extracted tree-adjoining grammar, *Proc. of the 9th RIVF*, pp.1-6, 2012.
[6] L. N. Thi, L. H. My, H. N. Viet, H. N. T. Minh and P. L. Hong, Building a treebank for Vietnamese dependency parsing, *Proc. of the 10th RIVF*, 2013.
[7] D. Q. Nguyen, D. Q. Nguyen, S. B. Pham, P.-T. Nguyen and M. L. Nguyen, From treebank conversion to automatic dependency parsing for Vietnamese, *Proc. of NLDB*, pp.196-207, 2014.
[8] D. Q. Nguyen and A. T. Nguyen, PhoBERT: Pre-trained language models for Vietnamese, *Findings of EMNLP 2020*, pp.1037-1042, 2020.
[9] L. T. Nguyen and D. Q. Nguyen, PhoNLP: A joint multi-task learning model for Vietnamese part-of-speech tagging, named entity recognition and dependency parsing, *Proc. of NAACL*, 2021.
[10] K. Clark, M.-T. Luong, C. D. Manning and Q. Le, Semi-supervised sequence modeling with cross-view training, *EMNLP*, pp.1914-1925, 2018.
[11] Y. Kim, Y. Jernite, D. Sontag and A. M. Rush, Character-aware neural language models, *Proc. of the 30th AAAI*, pp.2741-2749, 2016.
[12] *CLC_VN_POS*, http://www.clc.hcmus.edu.vn/?page_id=36, 2019.
[13] J. Devlin, M.-W. Chang, K. Lee and K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, *Proc. of NAACL*, pp.4171-4186, 2019.
[14] K. Clark, U. Khandelwal, O. Levy and C. D. Manning, What does BERT look at? An analysis of BERT's attention, *Proc. of BlackboxNLP*, Florence, Italy, pp.276-286, 2019.
[15] P. Qi, T. Dozat, Y. Zhang and C. D. Manning, Universal dependency parsing from scratch, *Proc. of the CoNLL 2018 Shared Task*, pp.160-170, 2018.
[16] R. K. Srivastava, K. Greff and J. Schmidhuber, Highway networks, *Proc. of ICML*, 2015.
[17] Y.-J. Chu and T.-H. Liu, On the shortest arborescence of a directed graph, *Science Sinica*, vol.14, pp.1396-1400, 1965.
[18] J. Edmonds, Optimum branchings, *J. Res. Natl. Bur. Stand.*, vol.71, pp.233-240, 1967.
[19] K. V. Nguyen and N. L.-T. Nguyen, Error analysis for Vietnamese dependency parsing, *Proc. of KSE*, pp.79-84, 2015.
[20] D. Q. Nguyen, M. Dras and M. Johnson, An empirical study for Vietnamese dependency parsing, *Proc. of ALTA*, pp.143-149, 2016.
[21] K. Mrini, F. Dernoncourt, T. Bui, W. Chang and N. Nakashole, Rethinking self-attention: Towards interpretability in neural parsing, *Findings of the Association for Computational Linguistics: EMNLP*, pp.731-742, 2020.

[22] J. Zhou, Z. Li and H. Zhao, Parsing all: Syntax and semantics, dependencies and spans, *Proc. of the 2020 Conference on EMNLP*, 2020.

[23] G. Jawahar, B. Sagot and D. Seddah, What does BERT learn about the structure of language?, *Proc. of the 57th Annual Meeting of the Association for Computational Linguistics*, pp.3651-3657, 2019.

[24] D. Truong, D.-T. Vo and U. T. Nguyen, Vietnamese open information extraction, *Proc. of SoICT*, pp.135-142, 2017.

[25] V. H. Tran, H. T. Vu, T. H. Pham, V. V. Nguyen and M. L. Nguyen, A reordering model for Vietnamese-English statistical machine translation using dependency information, *Proc. of RIVF*, pp.125-130, 2016.