Using Stanford neural dependency parser for Vietnamese dependency parsing

Nguyen Thi Thuy Lien
FPT University
Hanoi, Vietnam
thuylien2711@gmail.com

Pham Quang Nhat Minh
Aimesoft JSC
Hanoi, Vietnam
minhpham@aimesoft.com

***Abstract*—This paper presents our approach to the Vietnamese dependency parsing task in VLSP 2019 Evaluation Campaign. We used Stanford graph-based neural dependency parser [1], [2] and made a few modifications to adapt for Vietnamese dependency parsing. In evaluation on the private datasets, on average our best model obtained 70.75% UAS score and 58.75% LAS score.**

***Index Terms*—dependency parsing, Vietnamese, Stanford neural dependency parser.**

I. Introduction

Dependency parsing is an important component in many natural language processing (NLP) systems such as semantic role labeling [3] or relation extraction [4]. Given a sentence consisting of a sequence of tokens, the task is to identify in a sentence, pairs of a dependent token and a head token which have dependency relation and their dependency relation labels.

In VLSP 2019 evaluation campaign, dependency parsing shared task was proposed with the purpose of promoting the development of dependency parsers for Vietnamese. In this paper, we applied Stanford graph-based neural dependency parser [1], [2] which is the state-of-the-art universal dependency parser to Vietnamese dependency parsing task and conducted experiments on Vietnamese dataset provided by VLSP 2019 organizers. On the private test data, we obtained 70.75% UAS score and 58.75% LAS score (averaged on three test sets). The scores are relatively low compared with those for English language. It suggests that there will be much room for improvement in Vietnamese dependency parsing.

In the following sections, we briefly describe Stanford neural dependency parser [1], [2], and then present our experimental setup for Vietnamese dependency parsing. In Section III, we present experimental results obtained on the public and private test datasets. In Section IV, we give conclusions for the paper.

II. Methodology

A. Stanford neural dependency parser

(Dozat et al., 2017) [2] proposed a graph-based dependency parser using a deep biaffine neural network. More specifically, embedding vectors of words in a sentence are fed to a bidirectional LSTM network to generate word representations. The output state of the final LSTM layer is fed to four separate ReLU layers to produce four vector representations. Those vectors are used as input to two deep biaffine classifiers to predict dependent's heads and their relation labels.

TABLE I
Optimizers' hyper-parameters in three submitted models

	Hyper-parameters
Model 1	SGD optimizer; learning rate=0.02, momentum=0
Model 2	Adam optimizer; learning rate=0.001
Model 3	Adam optimizer; learning rate=0.003

B. Experimental settings for Vietnamese dependency parsing

In the paper, we used Stanford neural dependency parsing model to develop a dependency parser for Vietnamese language. We used the same hyperparameter configuration proposed by Dozat et al., 2017 [2] except a few modifications.

We investigated three models using different hyperparameters for the optimizer in training. Table I summarises optimizers' hyper-parameters used in three models. In the first model, we used the SGD optimizer [5] with the learning rate 0.02 and momentum value 0. The second model uses Adam optimizer [6] with the learning rate 0.001. The third model uses Adam optimizer with the learning rate 0.003.

Three models are trained on Vietnamese dependency treebank provided by VLSP 2019 organizers. The corpus contains 3,000 sentences and 43,754 tokens, which consist of a modest 14% (6,068) of tokens that are not followed by a space, 3,346 types of words with spaces, and 13 types of words accommodate of both letters and punctuation.

For the pre-trained word vectors, the one trained on Wikipedia using fastText [7] for Vietnamese has been chosen. The dimension of each word vector in fastText pre-trained word vectors is 300.

Each model was trained with up to 10,000 steps, where in one step, a minibatch with approximately 5,000 tokens was used. We saved a model checkpoint after 100 steps. Afterward, a new model checkpoint is only saved if validation accuracy increases. The training process is terminated if after 1,000 training steps, we do not obtain any improvement in the validation accuracy.

III. Experiments and Results

Evaluation metrics are the labeled attachment score (LAS) and unlabeled attachment score (UAS). LAS is the percentage

TABLE II
RESULTS ON THE PUBLIC TEST-SET

	UAS (%)	**LAS (%)**
Model 1	72.23	60.03
Model 2	74.87	62.16
Model 3	75.61	64.22

TABLE III
RESULTS ON THE PRIVATE TEST-SET

		VTB	**MXH**	**HTB**	**Avg. Score**
Model 1	UAS(%)	65.15	63.16	77.00	66.03
	LAS(%)	52.36	51.81	65.73	53.50
Model 2	UAS(%)	68.23	66.23	78.45	68.95
	LAS(%)	54.99	54.53	68.55	56.16
Model 3	UAS(%)	69.93	67.63	81.55	70.75
	LAS(%)	57.46	56.67	72.64	58.75

of tokens which have been correctly assigned heads and dependency relations, meanwhile, UAS is the proportion of nodes whose heads has been correctly assigned.

Table II show evaluation results of three models on the public test-set. Accordingly, the first model gives the lowest result with approximately 60% of LAS and 72.23% of UAS. In contrast to the first model, the third one reaches the best score in which LAS increases over 3% to 75.61% and UAS rises to 64.22%.

The private test data includes three data sets, VTB, MXH, and HTB. VTB is the data obtained from Viet Treebank. HTB is the data which was built by annotating on the Vietnamese translation of the story "The Little Prince" while MXH is annotated on social textual media data and online reviews.

Table III presents the evaluation results on these private sets corresponding to three submitted models. The results on the private test sets is consistent with the results on the public test. That is the third model obtained the highest scores among three models with 70.75% averaged UAS score and 58.75% averaged LAS score. The evaluation results also indicated that the scores on MXH test set are consistent lower than scores on VTB and HTB test sets. A plausible explanation is that the domain of MXH test set is different from the domain of the training data which is constructed from news articles.

IV. CONCLUSIONS

This paper presents our experiments of applying Stanford graph-based neural dependency parser for the Vietnamese dependency parsing task at VLSP 2019 Evaluation Campaign. We investigated three variant models with different optimization hyper-parameters and obtained promising results on the private test-set. However, the results are much lower than those for English language. It implies that there will probably still be room for improvement.

REFERENCES

[1] P. Qi, T. Dozat, Y. Zhang, and C. D. Manning, "Universal dependency parsing from scratch," in *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*. Brussels, Belgium: Association for Computational Linguistics, October 2018, pp. 160–170. [Online]. Available: https://nlp.stanford.edu/pubs/qi2018universal.pdf

[2] T. Dozat, P. Qi, and C. D. Manning, "Stanford's graph-based neural dependency parser at the conll 2017 shared task," in *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, 2017, pp. 20–30.

[3] D. Marcheggiani and I. Titov, "Encoding sentences with graph convolutional networks for semantic role labeling," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Copenhagen, Denmark: Association for Computational Linguistics, Sep. 2017, pp. 1506–1515.

[4] Y. Zhang, P. Qi, and C. D. Manning, "Graph convolution over pruned dependency trees improves relation extraction," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, Oct.-Nov. 2018, pp. 2205–2215.

[5] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," in *International conference on machine learning*, 2013, pp. 1139–1147.

[6] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[7] A. Joulin, E. Grave, P. Bojanowski, M. Douze, H. Jégou, and T. Mikolov, "Fasttext.zip: Compressing text classification models," *arXiv preprint arXiv:1612.03651*, 2016.