VLSP 2020 Shared Task: Universal Dependency Parsing for Vietnamese

HA My Linh[1]**, NGUYEN Thi Minh Huyen**[1]**, VU Xuan Luong**[2]**,**
NGUYEN Thi Luong[3]**, PHAN Thi Hue**[4]**, LE Van Cuong**[5]

[1]VNU University of Science, Hanoi, Vietnam
{huyenntm, hamylinh}@hus.edu.vn
[2]Vietnam Lexicography Center, Hanoi, Vietnam
vuluong@vietlex.com
[3]Dalat University, Lamdong, Vietnam
luongnt@dlu.edu.vn
[4]Coc Coc Company, Hanoi, Vietnam
phanhuek51nn@gmail.com
[5]University of Social Sciences and Humanities, Hanoi, Vietnam
cuongle.ussh@gmail.com

Abstract

This paper describes the shared task on Vietnamese universal dependency parsing at the seventh workshop on Vietnamese Language and Speech Processing (VLSP 2020[1]). This challenge, following the first edition in 2019, aims to provide the VLSP community with gold universal dependency annotated datasets for Vietnamese and to evaluate dependency parsing systems based on the same training and test sets. Consequently, the best systems made available to the community would be promoted for using in further applications. Each participant was provided with the same training data with more than 8000 annotated sentences and returned the result on a test set of more than 1000 sentences. Contrary to the first edition, where the test set was preprocessed with word segmentation and part-of-speech (POS) tagging in CoNLL-U format, participants of this year compete on two tracks: one track with raw texts and the other with preprocessed texts as test input. In this report, we define the shared task and describe data preparation, as well as make an overview of methods and results performed by VLSP 2020 participants.

1 Introduction

Dependency parsing is the task of determining syntactic dependencies between words in a sentence. The dependencies include, for example, the information about the relationship between a predicate and its arguments, or between a word and its modifiers. Dependency parsing can be applied in many tasks of natural language processing such as information extraction, co-reference resolution, question-answering, semantic parsing, etc.

Many shared-tasks on dependency parsing have been organized since 2006 by CoNLL (The SIGNLL Conference on Computational Natural Language Learning), not only for English but also for many other languages in a multilingual framework. The CoNLL 2017 Shared Task was done on 81 test sets from 49 languages and in CoNLL 2018 Shared Task (Zeman et al., 2018), there were 82 test sets from 57 languages. From 2017, a Vietnamese dependency treebank containing 3,000 sentences is included for the CoNLL shared-task "Multilingual Parsing from Raw Text to Universal Dependencies". However, this Vietnamese dependency treebank is still small and contains several errors because of automatic conversion from the version 1 to version 2 of Universal Dependencies[2] (UD v2).

In the framework of the VLSP 2019 and 2020 workshops, one of the shared-tasks is on Vietnamese dependency parsing, in order to promote the development of dependency parsers for Vietnamese. Based on newly revised guidelines for Vietnamese dependency treebank following the UD v2 annotation scheme, training and test sets have been annotated. The label set and guidelines on word segmentation and POS tagging were equally revised, in agreement with the universal principles. In 2020, participants are provided with more than 8,000 sentences for the training dataset. The test set includes more than 1000 sentences provided in two formats as two tracks of the challenge: one is raw text and the other is text segmented in words and POS tagged. The tool provided for evaluating

[1] https://vlsp.org.vn/vlsp2020

[2] https://universaldependencies.org/v2/index.html

dependency parsing models by the CoNLL 2018 shared task is used in the framework of VLSP 2019 and 2020 dependency parsing shared task.

Five participant systems have been evaluated in VLSP 2020. After the description of the datasets and evaluation methods, we give an overview of models developed by participant systems and discuss the results obtained by these systems on the two tracks of the shared tasks.

2 Data preparation

Training and test datasets have been automatically generated by a draft parsing system and manually revised by annotators. We introduce the set of dependency labels first, then the annotation process and finally the datasets built for the shared task of dependency parsing.

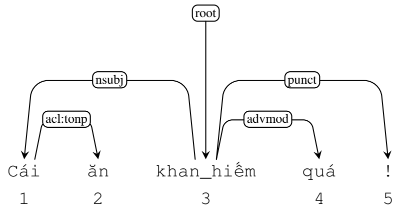
2.1 Dependency labels

In 2017, the NLP group of the VNU University of Science (Nguyen et al., 2018) has developed a Vietnamese dependency dataset of 3,000 sentences which were then integrated into Stanford University's dependency project. The label set is composed of 48 dependency labels, defined based on Universal Dependency label set (version 1). The 3,000 sentences of this dataset are extracted from VietTreebank - a constituency treebank, then automatically transformed into a dependency treebank. The process is terminated by a manual revision, although there exist inevitably some errors from inexperienced annotators. The UD v2 version of this dataset in Universal Dependency repositories was automatically generated from the version 1, it contains consequently much more errors.

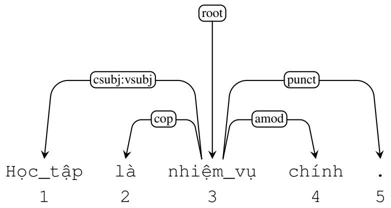
For the dependency shared task organized in the framework of VLSP 2019 and VLSP 2020 workshops, we have reviewed entirely the set of dependency labels and defined a set of 38 types and 47 language-specific subtypes of dependency relations in accordance with the guidelines for Universal dependency relations[3] version 2.

Here are some new dependency labels specific for Vietnamese language.

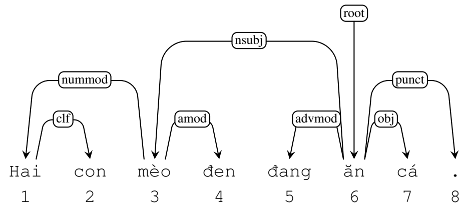
- *acl:tonp*: Usually a verb in Vietnamese can be nominalized by adding a classifier noun such as *cái*, *việc*, *sự*, ... before the verb. Example: *Cái*[classfier] *ăn*[to eat] *khan hiếm*[scarce] *quá*[too]!/The food is too scarce!



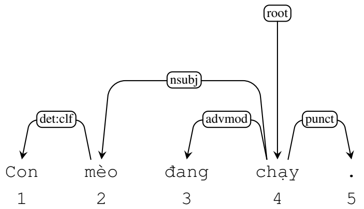
- *csubj:vsubj*: This relationship is used in the case that a verb is the subject of a sentence. In Vietnamese, the subject is usually nominal, but in some cases, adjective or verb without any derivation can be subject. Example: *Học tập*[Study] *là*[to be] *nhiệm vụ*[task] *chính*[main]./Studying is the main task.



- *clf*: Syntactically, classifier noun is rather bound to the numeral in the nominal group than the head noun. Therefore, in agreement with the guidelines of UD version 2, we treat classifiers as functional dependents of numerals, using the *clf* relation. Example: *Hai*[Two] *con*[classifer] *mèo*[cat] *đen*[black] *đang*[tense/aspect marker] *ăn*[to eat] *cá*[fish]./Two black cats are eating fish.



- *det:clf*: When a classifier noun does not appear with a specified quantity or pronoun words, the classifier noun have the same properties as a determiner. The relation of a classifier noun with the head noun is thus *det:clf*. Example: *Con*[Classifier] *mèo*[cat] *đang*[tense/aspect marker] *chạy*[to run]./The cat is running.



[3] https://universaldependencies.org/u/dep/index.html

Other relations, including *nsubj:nn*, *obl:tmod*, could be consulted in the dependency annotation guidelines released on the web of VLSP.

Regarding multiword expressions (MWEs), we have defined 16 subtypes for capturing different cases of MWEs in Vietnamese.

Dependency labels are described in detail in the guidelines published along with training data. Each relation is accompanied by a definition, examples and notes on ambiguous cases.

Besides the dependency labels, we also map Vietnamese POS tagset to Universal POS tagset. This work is important for the integration of Vietnamese dependency corpus to Universal dependency project. Guidelines for Vietnamese word segmentation and POS tagging used for VietTreebank published in 2009 have equally been revised, and the corpus published for the dependency shared task is annotated in accordance with these guidelines.

2.2 Annotation process

For an easy annotation of dependency relations, we have designed a tool exceptionally for this task.

The data annotation is performed by two linguists, one computer scientist and approved by one linguistic annotator expert. Finally, annotators cross-checked labeling results and discussed among them for obtaining the most accurate annotation.

Table 1 shows the inter-annotator agreement between each couple of annotators.

Table 1: Agreement between three annotators

Agreement	**UAS**	**LAS**
Ano1-Ano2	96.28	92.74
Ano1-Ano3	94.44	89.98
Ano2-Ano3	95.55	92.53
Average	95.42	91.75

2.3 Datasets

In 2019, the datasets are collected from three sources: 4000 sentences in VietTreebank corpus (articles crawled from the "Tuổi trẻ" news website), "Little Prince" corpus (a famous French novella, translated in hundreds of languages around the world), and a set of hotel and restaurant reviews (social network data).

All the training and test datasets from VLSP 2019 are provided as training data for VLSP 2020 shared task, in addition to about 4000 sentences from VietTreebank newly annotated in 2020. In total, the training set contains 8,152 sentences. The test data is composed of two sets: 906 sentences from VietTreebank and 217 sentences randomly collected from *VnExpress*[4].

For VLSP 2019, participants worked only with pre-processed datasets: all the sentences in the training and test set are segmented and POS tagged. In VLSP 2020, participants competed on two tracks: one with raw data and the other with data already segmented in words and POS tagged. At the first step, all the teams received the raw data and had one-day deadline to submit their result. At the second step, participants have been sent the same test set with word segmentation and POS tagging.

Table 2 gives some statistics on the datasets: the number of sentences and the average number of words per sentence.

Table 2: Number of sentences and average number of words per sentence

Data	**Number of Sentences**	**Length <30**	**Length 30-50**	**Length >50**	**Length Average**
Training Package1	5069	4882	159	28	14.40
Training Package2	3083	1942	1005	136	24.96
Test Data	1123	852	229	42	23.29

It can be seen that the sentences in the training dataset Package2 and testing data are much longer than sentences in Package1 from the previous year. This is not a small challenge for participants, because the longer the sentence is, the greater the complexity is. To tackle this problem, one needs to have smoother and more efficient pre-processing steps.

3 Parsing Methods

VLSP dependency parsing shared task counted 15 registered teams, but finally only 5 teams could submit results. All these teams (DP1, DP2, DP3, DP4 and DP5) actually deployed parsing models based on graph neural networks (Dozat et al., 2017), combining with different models of word embeddings.

3.1 Team DP1

The team DP1 proposed a joint deep contextualized word representation for dependency parsing. Their joint representation consists of five components: word representations from ELMo (Peters et al., 2018) and BERT (Devlin et al., 2019) language

[4] https://vnexpress.net/

models for Vietnamese (Nguyen and Tuan Nguyen, 2020), Word2Vec embeddings trained on Baomoi dataset (Xuan-Son Vu, 2019), character embeddings (Kim, 2014), and POS tag embeddings. This joint representation is finally deployed in a deep biaffine dependency parser (Dozat et al., 2017).

For raw data input, they used VnCoreNLP (Vu et al., 2018) for segmentation and POS tagging. A POS tag mapping was defined to convert from VnCoreNLP POS tagset into the universal tagset used in VLSP dependency data.

3.2 Team DP2

The team DP2 proposes a combining architecture of two state-of-the-art moels: PhoBERT - the Vietnamese language model (Nguyen and Tuan Nguyen, 2020), and the Biaffine Attention mechanism for universal dependency parsing.

For the encoder, they extract word vectors from two last layers of PhoBERT-base and concatenate them to form 1536-D word representations. The outputs of PhoBERT are passed through a word alignment layer to obtain aggregated word-based representations.

For decoding, they develop equally models for jointly learning POS taging and dependency parsing as proposed in (Nguyen and Verspoor, 2018). However, their experiments show that the best performance on their validation set obtained with the use of PhoBERT-large and biaffine attention mechanism without POS learning. The package is available on github[5].

VnCoreNLP (Vu et al., 2018) was used for preprocessing raw texts.

3.3 Team DP3

The team DP3 chose equally the model of Stanford's graph-based neural dependency parser to build their dependency parsing models. The team focused on testing four different configurations of embeddings: word embeddings or pre-trained word embeddings (Xuan-Son Vu, 2019) combined with character embeddings or with POS tag embeddings.

In case of raw data input, the team DP3 used underthesea[6] for word segmentation. For POS tagging, they have trained a POS tagger using bidirectional LSTM-CRF models for sequence tagging (Huang et al., 2015) and the same pre-trained word embeddings as above.

The results show that for raw text input, the use of character-level embeddings proves a better performance than POS tag embeddings. For CoNLL data input, using pre-trained word embeddings in combination with POS tag embeddings gives the best performance.

3.4 Team DP4

The solutions adopted by Team DP4 for building their parsing systems are as follows.

For dependency parsing, they implemented a BiLSTM-based deep biaffine neural dependency parser. They used Adam optimizer to optimize the network and fastText for word representations (Joulin et al., 2016). Two different models for dependency parsing have been built: the first uses both UPOS and XPOS information for training and predicting data, while the second uses only UPOS information during the entire process. Experiments show that the model using both UPOS and XPOS information generally gives better results.

For the preprocessing of raw data, VnCoreNLP (Vu et al., 2018) was used for sentence splitting and word segmentation. The POS tagging was performed by a BERT-based (Devlin et al., 2019) classifier using bertbase-multilingual-cased pretrained-model available in HuggingFace (Wolf et al., 2019).

3.5 Team DP5

The team DP5 uses Bidirectional Long Short-Term Memory (BiLSTM) (Kiperwasser and Goldberg, 2016) network to extract the contextual information, while the graph neural network captures high-order information. The pre-processing of raw texts, such as word segmentation and POS tagging, is performed by using VnCoreNLP (Vu et al., 2018).

For the word embedding layer, they adopted a pre-trained model for Vietnamese with 300-dimensional word embeddings, i.e. fastText (Joulin et al., 2016). Each word is embedded using three different vectors: randomly initialized word embedding, pre-trained word embedding, and POS embedding.

4 Evaluation

4.1 Data format

The dependency annotated texts are encoded in CoNLL-U format[7].

[5] https://github.com/quangph-1686a/VUDP

[6] https://pypi.org/project/underthesea/

[7] https://universaldependencies.org/format.html

Each sentence consists of one or more word lines, and each word line contains 10 fields as follows.

1. ID: Word index, integer starting at 1 for each new sentence.
2. FORM: Word form or punctuation symbol.
3. LEMMA: Lemma of the word, which is the same as the word form for Vietnamese.
4. UPOS: Universal POS tag. X if not available.
5. XPOS: Vietnamese POS tag; _ if not available.
6. FEATS: Morphological features; _ if not available.
7. HEAD: Head of the current word, which is either a value of ID or zero (0).
8. DEPREL: Universal dependency relation to the HEAD (root if HEAD = 0) or a defined language-specific sub-type of one.
9. DEPS: Enhanced dependency graph in the form of a list of head-deprel pairs.
10. MISC: Any other annotation.

An example is given in Table 3.

Table 3: A sentence in training set

1	Tôi	tôi	PROPN	Pro	_	3	nsubj	_	_
2	đã	đã	ADV	Adv	_	3	advmod	_	_
3	sống	sống	VERB	V	_	0	root	_	_
4	nhiều	nhiều	ADJ	Adj	_	3	advmod:adj	_	_
5	với	với	SCONJ	C	_	7	case	_	_
6	những	những	DET	Det	_	7	det	_	_
7	người lớn	người lớn	N	N	_	3	obl:with	_	_
8	.	.	PUNCT	PUNCT	_	3	punct	_	_

The 9th and 10th columns remain empty (_) in current datasets. For test data, the 7th and 8th columns are empty (_).

4.2 Evaluation metrics

VLSP 2020 participant systems are evaluated and ranked using the standard evaluation metric in dependency parsing which is Labeled Attachment Score (LAS), defined in comparing the gold relations of the test set and relations returned by the system:

$$P = \frac{correctRelations}{systemNodes}$$

$$R = \frac{correctRelations}{goldNodes}$$

$$LAS = \frac{2 * P * R}{(P + R)}$$

As in CoNLL 2018 dependency shared task (Zeman et al., 2018), for scoring purposes, only universal dependency labels will be taken into account, which means that language-specific subtypes such as acl:relcl (relative clause), a subtype of the universal relation acl (clausal modifier of noun), will be truncated to acl both in the gold standard and in the parser output in the evaluation.

In addition, UAS (Unlabeled Attachment Score) metric is also provided, showing the percentage of words that are assigned the correct syntactic head. We use the evaluation script published at CoNLL 2018[8].

4.3 Results

Table 4 shows the results obtained on raw text input of each system. The teams DP1 and DP3 submitted results of multiple models. Results from both the UAS and LAS measurements show the uniformity of the teams' models across all different data sets. Last two columns show the results of each team on the whole test set with the best systems highlighted.

Table 5 shows the results for the segmented and POS tagged text input in CoNLL-U format. More models have been submitted for this track. It can be seen that the results with this format are significantly higher than the raw data input, which is quite understandable, especially as the pre-processing tools are in agreement with older guidelines of word segmentation and POS tagging. An interesting observation is that the team DP2 achieves the first rank for raw text input but only the third rank for this pre-processed input: the model submitted by this team is the only model that doesn't use POS information. A possible interpretation is that erroneous POS labels had a strong negative impact on the results.

A statistic shows that all teams share a high intersection of 55.49% lines with the gold test dataset. An analysis in detail in the future would help us to understand better the characteristics of these common results.

Table 6 gives a closer look of the results regarding the sentence length. For all models, the accuracy decreases as the sentence length increases. This confirms the bigger challenge of the VLSP 2020 dependency parsing shared task in comparison with the task in VLSP 2019. In addition, given

[8] https://universaldependencies.org/conll18/conll18_ud_eval.py)

Table 4: Input: Raw text

Team	Model	VTB		vnexpress1		vnexpress3		vnexpress7		vnexpress8		vnexpress10		vnexpress14		Total	
		UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS
DP1	1	76.33	67.46	74.79	65.38	74.22	66.73	68.33	61.67	74.81	65.71	80.64	72.46	72.61	62.45	**76.12**	**67.32**
	2	75.68	66.59	72.17	62.61	74.95	67.28	66.11	61.11	74.29	65.97	78.45	69.98	73.36	63.69	**75.48**	**66.53**
DP2	1	78.49	68.94	79.72	70.62	78.37	70.08	68.89	65.56	78.31	70.00	81.08	74.80	74.85	68.15	**78.45**	**69.21**
DP3	1	76.44	67.68	73.05	63.78	75.91	68.05	66.67	64.44	62.52	55.86	73.36	67.16	68.16	61.19	**75.63**	**67.12**
	2	74.97	65.50	69.65	59.00	75.00	67.74	61.11	58.33	60.60	51.63	70.85	62.73	70.90	63.93	**74.15**	**64.93**
DP4	1	74.55	65.34	71.70	58.91	77.09	69.29	70.56	65.56	69.61	59.35	76.41	68.22	71.62	63.69	**74.47**	**65.3**
DP5	1	73.18	64.66	68.77	58.75	74.1	65.81	61.67	55.56	68.96	61.43	73.19	64.13	68.4	60.72	**72.85**	**64.35**

Table 5: Input: CoNLLU

	Model	VTB		vnexpress1		vnexpress3		vnexpress7		vnexpress8		vnexpress10		vnexpress14		Total	
		UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS
DP1	1	84.81	76.44	78.98	70.94	85.89	76.97	82.22	75.56	82.49	73.93	85.46	77.53	84.04	75.31	**84.65**	**76.27**
	2	84.58	76.29	77.43	70.17	85.46	77.58	80.00	73.89	81.32	73.80	81.20	72.69	83.54	76.81	**84.23**	**76.05**
DP2	1	83.36	73.29	82.84	73.42	83.81	74.59	81.11	75.00	80.67	72.11	85.76	78.85	81.80	73.32	**83.32**	**73.5**
DP3	1	80.12	70.71	75.73	66.15	80.15	71.72	74.44	70.56	76.01	66.28	82.09	74.74	76.81	69.58	**79.86**	**70.62**
	2	81.89	73.71	67.70	57.96	78.68	70.98	69.44	61.67	74.97	66.54	76.36	69.02	75.81	68.58	**80.81**	**72.66**
	3	80.81	71.71	76.20	67.23	79.47	71.29	74.44	70.00	76.26	68.09	82.09	74.16	79.05	71.07	**80.44**	**71.5**
	4	82.11	73.47	73.88	65.84	80.82	72.02	70.00	64.44	76.39	69.26	81.64	71.95	80.55	73.32	**81.53**	**72.96**
DP4	1	84.41	75.94	74.34	63.37	84.42	76.48	85.56	78.89	82.88	74.84	83.55	75.48	82.79	76.31	**84.08**	**75.64**
	2	83.20	75.14	68.32	55.95	76.60	68.48	71.11	61.11	70.56	61.09	73.13	63.29	75.56	68.08	**81.58**	**73.32**
DP5	1	81.89	73.34	75.12	66.15	84.36	75.38	76.67	67.22	79.25	71.98	80.47	72.54	80.55	73.57	**81.71**	**73.19**

Table 6: Statistics by the length of sentence

CoNLLU	< 30		30-50		> 50	
	UAS	LAS	UAS	LAS	UAS	LAS
DP1	86.11	77.37	82.75	75.12	80.89	72.61
DP2	84.28	74.52	81.90	72.49	81.34	69.87
DP3	82.61	74.01	80.10	71.62	78.88	70.23
DP4	84.83	76.15	83.05	75.08	82.30	74.05
DP5	83.06	74.36	79.81	71.89	78.60	69.47

Table 7: The final rank

No.	UAS	LAS	Aver.	Rank
DP1	80.39	71.80	76.09	2
DP2	80.89	71.36	76.12	1
DP3	78.58	70.04	74.31	4
DP4	79.28	70.47	74.87	3
DP5	77.28	68.77	73.03	5

the best model in 2019 obtained a performance of 73.53% for UAS and 61.28% for LAS, we can hope for improvement of all systems by enlarging the training dataset.

The teams are finally ranked based on the average of the best models for 2 testing data formats, as shown in Table 7.

5 Conclusion

We have presented the VLSP 2020 shared task on Dependency Parsing for Vietnamese. Although the number of registered participants for receiving the training datasets is 15, only 5 teams could submit the results. The other teams may not have enough time for achieving a satisfactory result, as many teams registered for several shared tasks at VLSP 2020. This shared task provides useful resources for building Vietnamese dependency parser and other applications that use dependency parsing results. We will continue to improve the quantity and quality of annotated sentences in order to get better performance in dependency parsing systems.

Acknowledgement

This shared task was supported by VINIF and VNG Zalo, as well as the NLP group at VNU University of Science.

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Timothy Dozat, Peng Qi, and Christopher D. Manning. 2017. 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*, pages 20–30, Vancouver, Canada. Association for Computational Linguistics.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional LSTM-CRF models for sequence tagging. *CoRR*, abs/1508.01991.

Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov. 2016. Fasttext.zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar. Association for Computational Linguistics.

Eliyahu Kiperwasser and Yoav Goldberg. 2016. Simple and accurate dependency parsing using bidirectional LSTM feature representations. *Transactions of the Association for Computational Linguistics*, 4:313–327.

Dat Quoc Nguyen and Anh Tuan Nguyen. 2020. PhoBERT: Pre-trained language models for Vietnamese. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1037–1042, Online. Association for Computational Linguistics.

Dat Quoc Nguyen and Karin Verspoor. 2018. An improved neural network model for joint POS tagging and dependency parsing. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 81–91, Brussels, Belgium. Association for Computational Linguistics.

Thi Luong Nguyen, My Linh Ha, Nguyên Thi Minh Huyên, and Phuong Le-Hong. 2018. Using BiLSTM in Dependency Parsing for Vietnamese. *Computación y Sistemas*, 22(3).

M. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Thanh Vu, Dat Quoc Nguyen, Dai Quoc Nguyen, Mark Dras, and Mark Johnson. 2018. VnCoreNLP: A Vietnamese natural language processing toolkit. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 56–60, New Orleans, Louisiana. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *CoRR*, abs/1910.03771.

Son N. Tran Lili Jiang Xuan-Son Vu, Thanh Vu. 2019. Etnlp: A visual-aided systematic approach to select pre-trained embeddings for a downstream task. *In: Proceedings of the International Conference Recent Advances in Natural Language Processing (RANLP)*.

Daniel Zeman, Jan Hajič, Martin Popel, Martin Potthast, Milan Straka, Filip Ginter, Joakim Nivre, and Slav Petrov. 2018. CoNLL 2018 shared task: Multilingual parsing from raw text to universal dependencies. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 1–21, Brussels, Belgium. Association for Computational Linguistics.