A neural joint model for Vietnamese word segmentation, POS tagging and dependency parsing

Dat Quoc Nguyen[1,2]
[1]The University of Melbourne, Australia
dqnguyen@unimelb.edu.au
[2]VinAI Research, Hanoi, Vietnam
v.datnq9@vinai.io

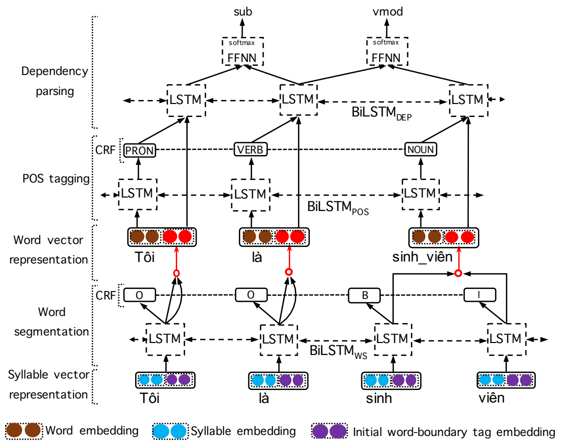
Abstract

We propose the first multi-task learning model for joint Vietnamese word segmentation, part-of-speech (POS) tagging and dependency parsing. In particular, our model extends the BIST graph-based dependency parser (Kiperwasser and Goldberg, 2016) with BiLSTM-CRF-based neural layers (Huang et al., 2015) for word segmentation and POS tagging. On Vietnamese benchmark datasets, experimental results show that our joint model obtains state-of-the-art or competitive performances.

1 Introduction

Dependency parsing (Kübler et al., 2009) is extremely useful in many downstream applications such as relation extraction (Bunescu and Mooney, 2005) and machine translation (Galley and Manning, 2009). POS tags are essential features used in dependency parsing. In real-world parsing, most parsers are used in a pipeline process with a precursor POS tagging model for producing predicted POS tags. In English where white space is a strong word boundary indicator, POS tagging is considered to be the first important step towards dependency parsing (Ballesteros et al., 2015).

Unlike English, for Vietnamese NLP, word segmentation is considered to be the key first step. This is because when written, white space is used in Vietnamese to separate syllables that constitute words, in addition to marking word boundaries (Nguyen et al., 2009). For example, a 4-syllable written text “Tôi là sinh viên” (I am student) forms 3 words “Tôi$_{\text{I}}$ là$_{\text{am}}$ sinh_viên$_{\text{student}}$”.[1] When parsing real-world Vietnamese text where gold word segmentation is not available, a pipeline process is defined that starts with a word segmenter to segment the text. The segmented text



ID	Form	POS	Head	DepRel
1	Tôi $_{\text{I}}$	PRON	2	sub
2	là $_{\text{am}}$	VERB	0	root
3	sinh_viên $_{\text{student}}$	NOUN	2	vmod

Figure 1: Illustration of our joint model. Linear transformations are not shown for simplification.

(e.g. “Tôi là sinh_viên”) is provided as the input to the POS tagger, which automatically generates POS-annotated text (e.g. “Tôi/PRON là/VERB sinh_viên/NOUN”) which is in turn fed to the parser. See Figure 1 for the final parsing output.

However, Vietnamese word segmenters and POS taggers have a non-trivial error rate, thus leading to error propagation. A solution to these problems is to develop models for jointly learning word segmentation, POS tagging and dependency parsing, such as those that have been actively explored for Chinese. These include traditional feature-based models (Hatori et al., 2012; Qian and Liu, 2012; Zhang et al., 2014, 2015) and neural models (Kurita et al., 2017; Li et al., 2018). These models construct *transition*-based frameworks at character level.

In this paper, we present a new multi-task learning model for joint word segmentation, POS tagging and dependency parsing. More specifically, our model can be viewed as an extension of the BIST *graph*-based dependency parser (Kiper-

[1]About 85% of Vietnamese word types are composed of at least two syllables and 80%+ of syllable types are words by themselves (Thang et al., 2008). For Vietnamese word segmentation, white space is only used to separate word tokens while underscore is used to separate syllables inside a word.

wasser and Goldberg, 2016), that incorporates BiLSTM-CRF-based architectures (Huang et al., 2015) to predict the segmentation and POS tags. To the best of our knowledge, our model is the first one which is proposed to jointly learn these three tasks for Vietnamese. Experiments on Vietnamese benchmark datasets show that our model produces state-of-the-art or competitive results.

2 Our proposed model

As illustrated in Figure 1, our joint multi-task model can be viewed as a hierarchical mixture of three components: word segmentation, POS tagging and dependency parsing. In particular, our word segmentation component formalizes the Vietnamese word segmentation task as a sequence labeling problem, thus uses a BiLSTM-CRF architecture (Huang et al., 2015) to predict BIO word boundary tags from input syllables, resulting in a word-segmented sequence. As for word segmentation, our POS tagging component also uses a BiLSTM-CRF to predict POS tags from the sequence of segmented words. Based on the input segmented words and their predicted POS tags, our dependency parsing component uses a graph-based architecture similarly to the one from Kiperwasser and Goldberg (2016) to decode dependency arcs and labels.

Syllable vector representation: Given an input sentence $\mathcal{S}$ of m syllables s_1, s_2, ..., s_m, we apply an initial word segmenter to produce initial BIO word-boundary tags b_1, b_2, ..., b_m. Following the state-of-the-art Vietnamese word segmenter VnCoreNLP's RDRsegmenter (Nguyen et al., 2018b), our initial word segmenter is based on the lexicon-based longest matching strategy (Poowarawan, 1986). We create a vector $\mathbf{v}_i$ to represent each i^{th} syllable in the input sentence $\mathcal{S}$ by concatenating its syllable embedding $\mathbf{e}^{(\mathrm{S})}_{s_i}$ and its initial word-boundary tag embedding $\mathbf{e}^{(\mathrm{B})}_{b_i}$:

$$\mathbf{v}_i = \mathbf{e}^{(\mathrm{S})}_{s_i} \circ \mathbf{e}^{(\mathrm{B})}_{b_i} \tag{1}$$

Word segmentation (WSeg): The WSeg component uses a BiLSTM (BiLSTM$_{\mathrm{WS}}$) to learn a latent feature vector representing the i^{th} syllable from a sequence of vectors $\mathbf{v}_{1:m}$:

$$\mathbf{r}^{(\mathrm{WS})}_i = \mathrm{BiLSTM}_{\mathrm{WS}}(\mathbf{v}_{1:m}, i) \tag{2}$$

The WSeg component then uses a single-layer feed-forward network (FFNN$_{\mathrm{WS}}$) to perform linear transformation over each latent feature vector:

$$\mathbf{h}^{(\mathrm{WS})}_i = \mathrm{FFNN}_{\mathrm{WS}}\left(\mathbf{r}^{(\mathrm{WS})}_i\right) \tag{3}$$

Next, the WSeg component feeds output vectors $\mathbf{h}^{(\mathrm{WS})}_i$ into a linear-chain CRF layer (Lafferty et al., 2001) for final BIO word-boundary tag prediction. A cross-entropy objective loss $\mathcal{L}_{\mathrm{WS}}$ is computed during training, while the Viterbi algorithm is used for decoding.

Word vector representation: Assume that we form n words w_1, w_2, ..., w_n based on m syllables in the input sentence $\mathcal{S}$. Note that we use gold word segmentation when training, and use predicted segmentation produced by the WSeg component when decoding. We create a vector $\mathbf{x}_j$ to represent each j^{th} word w_j by concatenating its word embedding $\mathbf{e}^{(\mathrm{W})}_{w_j}$ and its syllable-level word embedding $\mathbf{e}^{(\mathrm{SW})}_{w_j}$:

$$\mathbf{x}_j = \mathbf{e}^{(\mathrm{W})}_{w_j} \circ \mathbf{e}^{(\mathrm{SW})}_{w_j} \tag{4}$$

Here, inspired by Bohnet et al. (2018), to obtain $\mathbf{e}^{(\mathrm{SW})}_{w_j}$, we combine sentence-level context sensitive syllable encodings (from Equation 2) and feed it into a FFNN (FFNN$_{\mathrm{SW}}$):

$$\mathbf{e}^{(\mathrm{SW})}_{w_j} = \mathrm{FFNN}_{\mathrm{SW}}\left(\mathbf{r}^{(\mathrm{WS})}_{f(w_j)} \circ \mathbf{r}^{(\mathrm{WS})}_{l(w_j)}\right) \tag{5}$$

where $f(w_j)$ and $l(w_j)$ denote indices of the first and last syllables of w_j in $\mathcal{S}$, respectively.

POS tagging: The POS tagging component first feeds a sequence of vectors $\mathbf{x}_{1:n}$ into a BiLSTM (BiLSTM$_{\mathrm{POS}}$) to learn latent feature vectors representing input words, and passes each of these latent vectors as input to a FFNN (FFNN$_{\mathrm{POS}}$):

$$\mathbf{r}^{(\mathrm{POS})}_j = \mathrm{BiLSTM}_{\mathrm{POS}}(\mathbf{x}_{1:n}, j) \tag{6}$$

$$\mathbf{h}^{(\mathrm{POS})}_j = \mathrm{FFNN}_{\mathrm{POS}}\left(\mathbf{r}^{(\mathrm{POS})}_j\right) \tag{7}$$

Output vectors $\mathbf{h}^{(\mathrm{POS})}_j$ are then fed into a CRF layer for POS tag prediction. A cross-entropy loss $\mathcal{L}_{\mathrm{POS}}$ is computed for POS tagging when training.

Dependency parsing: Assume that the POS tagging component produces p_1, p_2, ..., p_n as predicted POS tags for the input words w_1, w_2, ..., w_n, respectively. Each j^{th} predicted POS tag p_j is represented by an embedding $\mathbf{e}^{(\mathrm{P})}_{p_j}$. We create a sequence of vectors $\mathbf{z}_{1:n}$ as input for the dependency parsing component, in which each $\mathbf{z}_j$ is resulted by concatenating the word vector representation $\mathbf{x}_j$ (from Equation 4) and the corresponding POS tag embedding $\mathbf{e}^{(\mathrm{P})}_{p_j}$. The dependency parsing component uses a BiLSTM ($\text{BiLSTM}_{\text{DEP}}$) to learn latent feature representations from the input $\mathbf{z}_{1:n}$:

$$\mathbf{z}_j = \mathbf{x}_j \circ \mathbf{e}^{(\text{P})}_{p_j} \quad (8)$$

$$\mathbf{r}^{(\text{DEP})}_j = \text{BiLSTM}_{\text{DEP}}(\mathbf{z}_{1:n}, j) \quad (9)$$

Based on latent feature vectors $\mathbf{r}^{(\text{DEP})}_j$, either a transition-based or graph-based neural architecture can be applied for dependency parsing (Kiperwasser and Goldberg, 2016).

Nguyen et al. (2016) show that in both neural network-based and traditional feature-based categories, graph-based parsers perform better than transition-based parsers for Vietnamese. Thus, our parsing component is constructed similarly to the BIST graph-based dependency parser from Kiperwasser and Goldberg (2016). A difference is that we use FFNNs to split $\mathbf{r}^{(\text{DEP})}_j$ into *head* and *dependent* representations:

$$\mathbf{h}^{(\text{A-H})}_j = \text{FFNN}_{\text{Arc-Head}}\left(\mathbf{r}^{(\text{DEP})}_j\right) \quad (10)$$

$$\mathbf{h}^{(\text{A-D})}_j = \text{FFNN}_{\text{Arc-Dep}}\left(\mathbf{r}^{(\text{DEP})}_j\right) \quad (11)$$

$$\mathbf{h}^{(\text{L-H})}_j = \text{FFNN}_{\text{Label-Head}}\left(\mathbf{r}^{(\text{DEP})}_j\right) \quad (12)$$

$$\mathbf{h}^{(\text{L-D})}_j = \text{FFNN}_{\text{Label-Dep}}\left(\mathbf{r}^{(\text{DEP})}_j\right) \quad (13)$$

To score a potential dependency arc, we use a FFNN (FFNN_{ARC}) with a one-node output layer:

$$\text{score}(i, j) = \text{FFNN}_{\text{ARC}}\left(\mathbf{h}^{(\text{A-H})}_i \circ \mathbf{h}^{(\text{A-D})}_j\right) \quad (14)$$

Given scores of word pairs, we predict the highest scoring projective parse tree by using the Eisner (1996) decoding algorithm. This unlabeled parsing model is trained with a margin-based hinge loss $\mathcal{L}_{\text{ARC}}$ (Kiperwasser and Goldberg, 2016).

To label predicted arcs, we use another FFNN ($\text{FFNN}_{\text{LABEL}}$) with softmax output:

$$\boldsymbol{v}_{(i,j)} = \text{FFNN}_{\text{LABEL}}\left(\mathbf{h}^{(\text{L-H})}_i \circ \mathbf{h}^{(\text{L-D})}_j\right) \quad (15)$$

Based on vectors $\boldsymbol{v}_{(i,j)}$, a cross entropy loss $\mathcal{L}_{\text{LABEL}}$ for dependency label prediction is computed when training, using the gold labeled tree.

Joint multi-task learning: We train our model by summing $\mathcal{L}_{\text{WS}}$, $\mathcal{L}_{\text{POS}}$, $\mathcal{L}_{\text{ARC}}$ and $\mathcal{L}_{\text{LABEL}}$ losses prior to computing gradients. Model parameters are learned to minimize the sum of the losses.

Discussion: Our model is inspired by stack propagation based methods (Zhang and Weiss, 2016; Hashimoto et al., 2017) which are joint models for POS tagging and dependency parsing. For dependency parsing, the Stack-propagation model (Zhang and Weiss, 2016) uses a transition-based approach, and the joint multi-task model JMT (Hashimoto et al., 2017) uses a head selection based approach which produces a probability distribution over possible heads for each word (Zhang et al., 2017), while our model uses a graph-based approach.

Our model can be viewed as an extension of the joint POS tagging and dependency parsing model jPTDP-v2 (Nguyen and Verspoor, 2018),[2] where we incorporate a BiLSTM-CRF for word boundary prediction. Other improvements to jPTDP-v2 include: (i) instead of using 'local' single word-based character-level embeddings, we use 'global' sentence-level context for learning word embeddings (see equations 2 and 5), (ii) we use a CRF layer for POS tagging instead of a softmax layer, and (iii) following Dozat and Manning (2017), we employ *head* and *dependent* projection representations (in Equations 10–13) as feature vectors for dependency parsing rather than the top recurrent states (in Equation 9).

3 Experimental setup

Datasets: We follow the setup used in the Vietnamese NLP toolkit VnCoreNLP (Vu et al., 2018).

For word segmentation and POS tagging, we use standard datasets from the Vietnamese Language and Speech Processing (VLSP) 2013 shared tasks.[3] To train the word segmentation layer, we use 75K manually word-segmented sentences in which 70K sentences are used for training and 5K sentences are used for development. For POS tagging, we use 27,870 manually word-segmented and POS-annotated sentences in which 27K and 870 sentences are used for training and development, respectively. For both tasks, the test set consists of 2120 manually word-segmented and POS-annotated sentences.

To train the dependency parsing layer, we use the benchmark Vietnamese dependency treebank VnDT (v1.1) of 10,197 sentences (Nguyen et al., 2014), and follow a standard split to use 1,020 sentences for test, 200 sentences for development and the remaining 8,977 sentences for training.

Implementation: We implement our model (namely, **jointWPD**) using DyNet (Neubig et al.,

[2]On the benchmark English PTB-WSJ corpus, jPTDP-v2 does better than Stack-propagation, while obtaining similar performance to JMT.

[3]`http://vlsp.org.vn/vlsp2013`

	Model	WSeg	PTag	LAS	UAS
Unsegmented	Our jointWPD	97.81	94.05	71.50	77.23
	VnCoreNLP	97.90	94.06	68.84**	74.52**
	jPTDP-v2	97.90	93.82*	70.78**	76.80*
	Biaffine	97.90	94.06	72.59**	78.54**

Table 1: F_1 scores (in %) for word segmentation (WSeg), POS tagging (PTag) and dependency parsing (LAS and UAS) on *test* sets of unsegmented sentences. Scores are computed on all tokens (including punctuation), employing the CoNLL 2017 shared task evaluation script (Zeman et al., 2017). In all tables, $*$ and $**$ denote the statistically significant differences against jointWPD at $p \leq 0.05$ and $p \leq 0.01$, respectively. We compute sentence-level scores for each model and task, then use paired t-test to measure the significance level.

Model	WSeg	PTag	LAS	UAS
WS $\mapsto$ Pos $\mapsto$ Dep	98.48*	95.09*	70.68*	76.70*
Our jointWPD	98.66	95.35	71.13	77.01
(a) w/o Initial$_{\text{BIO}}$	98.25**	95.01*	70.34**	76.36**
(b) w/o CRF$_{\text{WSeg}}$	98.32**	95.06*	70.48**	76.47**
(c) w/o CRF$_{\text{PTag}}$	98.65	95.14*	71.00	76.94
(d) w/o PTag	98.63	95.10*	69.78**	76.03**

Table 2: F_1 scores on *development* sets of unsegmented sentences. (a): Without using initial word-boundary tag embedding, i.e., Equation 1 becomes $\mathbf{v}_i = \mathbf{e}^{(s)}_{s_i}$; (b): Using a softmax layer for word-boundary tag prediction instead of a CRF layer; (c): Using a softmax layer for POS tag prediction instead of a CRF layer; (d): Without using the POS tag embeddings for the parsing component, i.e. Equation 8 becomes $\mathbf{z}_j = \mathbf{x}_j$.

2017). We learn model parameters using Adam (Kingma and Ba, 2014), and run for 50 epochs. We compute the average of F_1 scores computed for word segmentation, POS tagging and (LAS) dependency parsing after each training epoch. We choose the model with the highest average score over the development sets to apply to the test sets. See Appendix for implementation details.

4 Main results

End-to-end results: Our scores on the test sets are presented in Table 1. We compare our scores with the VnCoreNLP toolkit (Vu et al., 2018) which produces the previous highest reported results on the same test sets for the three tasks. Note that published scores of VnCoreNLP for POS tagging and dependency parsing were reported using gold word segmentation, and its published scores for dependency parsing were reported using the previous VnDT v1.0. As the current released VnCoreNLP version is retrained using the VnDT v1.1 and we also use the same experimental setup, we thus rerun VnCoreNLP on the unsegmented test sentences and compute its scores.[4] Our jointWPD obtains a slightly lower word segmentation score and a similar POS tagging score against VnCoreNLP. However, jointWPD achieves 2.7% absolute higher LAS and UAS than VnCoreNLP.

We also show in Table 1 scores of the joint POS tagging and dependency parsing model jPTDP-v2 (Nguyen and Verspoor, 2018) and the state-of-the-art Biaffine dependency parser (Dozat and Manning, 2017). For Biaffine which requires automatically predicted POS tags, following Vu et al. (2018), we produce the predicted POS tags on the whole VnDT treebank by using VnCoreNLP. We train both jPTDP-v2 and Biaffine with gold word segmentation.[5] For test, these models are fed with predicted word-segmented test sentences produced by VnCoreNLP. Our jointWPD performs significantly better than jPTDP-v2 on both POS tagging and dependency parsing tasks. However, jointWPD obtains 1.1+% lower LAS and UAS than Biaffine which uses a "*biaffine*" attention mechanism for predicting dependency arcs and labels. We will extend our parsing component with the biaffine attention mechanism to investigate the benefit for our joint model in future work.

Ablation analysis: Table 2 shows performance of a Pipeline strategy WS $\mapsto$ Pos $\mapsto$ Dep where we treat our word segmentation, POS tagging and dependency parsing components as independent networks, and train them separately. We find that jointWPD does significantly better than the Pipeline strategy on all three tasks.

Table 2 also presents ablation tests over 4 factors. When not using either initial word-boundary tag embeddings or the CRF layer for word-boundary tag prediction, all scores degrade by about 0.3+% absolutely. The 2 remaining factors, including (c) using a softmax classifier for

[4]See accuracy results w.r.t. the gold word segmentation in Table 3 in the Appendix.

[5]We reimplement jPTDP-v2 such that its POS tagging layer makes use of the VLSP 2013 POS tagging training set of 27K sentences, and then perform hyper-parameter tuning. The original jPTDP-v2 implementation only uses gold POS tags available in 8,977 training dependency trees, thus giving lower parsing performance than ours. For Biaffine, we use its updated version (Dozat et al., 2017) which won the CoNLL 2017 shared task on multilingual Universal Dependencies (UD) parsing from raw text (Zeman et al., 2017). Biaffine was also employed in all the top systems at the follow-up CoNLL 2018 shared task (Zeman et al., 2018).

POS tag prediction rather than a CRF layer and (d) removing POS tag embeddings, do not effect the word segmentation score. Both factors notably decrease the POS tagging score. Factor (c) slightly decreases LAS and UAS parsing scores. Factor (d) degrades the parsing scores by about 1.0+%, clearly showing the usefulness of POS tag information for the dependency parsing task.

5 Related work

Nguyen et al. (2018b) propose a transformation rule-based learning model RDRsegmenter for Vietnamese word segmentation, which obtains the highest performance to date. Nguyen et al. (2017) briefly review word segmentation and POS tagging approaches for Vietnamese. In addition, Nguyen et al. (2017) also present an empirical comparison between state-of-the-art feature- and neural network-based models for Vietnamese POS tagging, and show that a conventional feature-based model performs better than neural network-based models. In particular, on the VLSP 2013 POS tagging dataset, MarMoT (Mueller et al., 2013) obtains better accuracy than BiLSTM-CRF-based models with LSTM- and CNN-based character level word embeddings (Lample et al., 2016; Ma and Hovy, 2016). Vu et al. (2018) incorporate RDRsegmenter and MarMoT as the word segmentation and POS tagging components of VnCoreNLP, respectively.

Thi et al. (2013) propose a conversion method to automatically convert the manually built Vietnamese constituency treebank (Nguyen et al., 2009) into a dependency treebank. However, Thi et al. (2013) do not clarify how dependency labels are inferred; also, they ignore syntactic information encoded in grammatical function tags, and unable to deal with coordination and empty category cases.[6] Nguyen et al. (2014) later present a new conversion method to tackle all those issues, producing the high quality dependency treebank VnDT which is then widely used in Vietnamese dependency parsing research (Nguyen and Nguyen, 2015, 2016; Nguyen et al., 2016, 2018a; Vu et al., 2018). Recently, Nguyen (2018) manually builds another Vietnamese dependency treebank—BKTreebank—consisting of about 7K sentences based on the Stanford Dependencies annotation scheme (Marneffe and Manning, 2008).

[6]Thi et al. (2013) reformed their dependency treebank with the UD annotation scheme to create a Vietnamese UD treebank in 2017. Note that the CoNLL 2017 & 2018 multilingual parsing shared tasks also provided F_1 scores for word segmentation, POS tagging and dependency parsing on this Vietnamese UD treebank. However, this UD treebank is small (containing about 1,400 training sentences), thus it might not be ideal to draw a reliable conclusion.

6 Conclusions and future work

In this paper, we have presented the first multi-task learning model for joint word segmentation, POS tagging and dependency parsing in Vietnamese. Experiments on Vietnamese benchmark datasets show that our joint multi-task model obtains results competitive with the state-of-the-art.

Che et al. (2018) show that deep contextualized word representations (Peters et al., 2018; Devlin et al., 2019) help improve the parsing performance. We will evaluate effects of the contextualized representations to our joint model. A Vietnamese syllable is analogous to a character in other languages such as Chinese and Japanese. Thus we will also evaluate the application of our model to those languages in future work.

Acknowledgments

I would like to thank Karin Verspoor and Vu Cong Duy Hoang as well as the anonymous reviewers for their feedback.

References

Miguel Ballesteros, Chris Dyer, and Noah A. Smith. 2015. Improved transition-based parsing by modeling characters instead of words with LSTMs. In *Proceedings of EMNLP*, pages 349–359.

Bernd Bohnet, Ryan McDonald, Gonçalo Simões, Daniel Andor, Emily Pitler, and Joshua Maynez. 2018. Morphosyntactic Tagging with a Meta-BiLSTM Model over Context Sensitive Token Encodings. In *Proceedings of ACL*, pages 2642–2652.

Razvan Bunescu and Raymond Mooney. 2005. A Shortest Path Dependency Kernel for Relation Extraction. In *Proceedings of HLT-EMNLP*, pages 724–731.

Wanxiang Che, Yijia Liu, Yuxuan Wang, Bo Zheng, and Ting Liu. 2018. Towards better UD parsing: Deep contextualized word embeddings, ensemble, and treebank concatenation. In *Proceedings of the CoNLL 2018 Shared Task*, pages 55–64.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of NAACL-HLT*.

Timothy Dozat and Christopher D. Manning. 2017. Deep Biaffine Attention for Neural Dependency Parsing. In *Proceedings of ICLR*.

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*, pages 20–30.

Jason M. Eisner. 1996. Three New Probabilistic Models for Dependency Parsing: An Exploration. In *Proceedings of COLING*, pages 340–345.

Michel Galley and Christopher D. Manning. 2009. Quadratic-Time Dependency Parsing for Machine Translation. In *Proceedings of ACL-IJCNLP*, pages 773–781.

Kazuma Hashimoto, Caiming Xiong, Yoshimasa Tsuruoka, and Richard Socher. 2017. A joint many-task model: Growing a neural network for multiple NLP tasks. In *Proceedings of EMNLP*, pages 1923–1933.

Jun Hatori, Takuya Matsuzaki, Yusuke Miyao, and Jun'ichi Tsujii. 2012. Incremental Joint Approach to Word Segmentation, POS Tagging, and Dependency Parsing in Chinese. In *Proceedings of ACL*, pages 1045–1053.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional LSTM-CRF Models for Sequence Tagging. *arXiv preprint*, arXiv:1508.01991.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *arXiv preprint*, arXiv:1412.6980.

Eliyahu Kiperwasser and Yoav Goldberg. 2016. Simple and Accurate Dependency Parsing Using Bidirectional LSTM Feature Representations. *Transactions of ACL*, 4:313–327.

Sandra Kübler, Ryan McDonald, and Joakim Nivre. 2009. *Dependency Parsing*. Synthesis Lectures on Human Language Technologies, Morgan & cLaypool publishers.

Shuhei Kurita, Daisuke Kawahara, and Sadao Kurohashi. 2017. Neural Joint Model for Transition-based Chinese Syntactic Analysis. In *Proceedings of ACL*, pages 1204–1214.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proceedings of ICML*, pages 282–289.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural Architectures for Named Entity Recognition. In *Proceedings of NAACL-HLT*, pages 260–270.

Haonan Li, Zhisong Zhang, Yuqi Ju, and Hai Zhao. 2018. Neural Character-level Dependency Parsing for Chinese. In *Proceedings of AAAI*, pages 5205–5212.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF. In *Proceedings of ACL*, pages 1064–1074.

Marie-catherine De Marneffe and Christopher D. Manning. 2008. The Stanford typed dependencies representation. In *Proceedings of the Coling 2008 workshop on Cross-Framework and Cross-Domain Parser Evaluation*, pages 1–8.

Thomas Mueller, Helmut Schmid, and Hinrich Schütze. 2013. Efficient higher-order CRFs for morphological tagging. In *Proceedings of EMNLP*, pages 322–332.

Graham Neubig, Chris Dyer, Yoav Goldberg, et al. 2017. DyNet: The Dynamic Neural Network Toolkit. *arXiv preprint*, arXiv:1701.03980.

Binh Duc Nguyen, Kiet Van Nguyen, and Ngan Luu-Thuy Nguyen. 2018a. LSTM Easy-first Dependency Parsing with Pre-trained Word Embeddings and Character-level Word Embeddings in Vietnamese. In *Proceedings of KSE*, pages 187–192.

Dat Quoc Nguyen, Mark Dras, and Mark Johnson. 2016. An empirical study for Vietnamese dependency parsing. In *Proceedings of ALTA*, pages 143–149.

Dat Quoc Nguyen, Dai Quoc Nguyen, Son Bao Pham, Phuong-Thai Nguyen, and Minh Le Nguyen. 2014. From Treebank Conversion to Automatic Dependency Parsing for Vietnamese. In *Proceedings of NLDB*, pages 196–207.

Dat Quoc Nguyen, Dai Quoc Nguyen, Thanh Vu, Mark Dras, and Mark Johnson. 2018b. A Fast and Accurate Vietnamese Word Segmenter. In *Proceedings of LREC*, pages 2582–2587.

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*, pages 81–91.

Dat Quoc Nguyen, Thanh Vu, Dai Quoc Nguyen, Mark Dras, and Mark Johnson. 2017. From Word Segmentation to POS Tagging for Vietnamese. In *Proceedings of ALTA*, pages 108–113.

Kiem-Hieu Nguyen. 2018. BKTreebank: Building a Vietnamese Dependency Treebank. In *Proceedings LREC*, pages 2164–2168.

Kiet Van Nguyen and Ngan Luu-Thuy Nguyen. 2015. Error Analysis for Vietnamese Dependency Parsing. In *Proceedings of KSE*, pages 79–84.

Kiet Van Nguyen and Ngan Luu-Thuy Nguyen. 2016. Vietnamese transition-based dependency parsing with supertag features. In *Proceedings of KSE*, pages 175–180.

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 LAW*, pages 182–185.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of NAACL-HLT*, pages 2227–2237.

Yuen Poowarawan. 1986. Dictionary-based Thai Syllable Separation. In *Proceedings of the Ninth Electronics Engineering Conference*, pages 409–418.

Xian Qian and Yang Liu. 2012. Joint Chinese Word Segmentation, POS Tagging and Parsing. In *Proceedings of EMNLP-CoNLL*, pages 501–511.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, 15:1929–1958.

Dinh Quang Thang, Le Hong Phuong, Nguyen Thi Minh Huyen, Nguyen Cam Tu, Mathias Rossignol, and Vu Xuan Luong. 2008. Word segmentation of Vietnamese texts: a comparison of approaches. In *Proceedings of LREC*, pages 1933–1936.

Luong Nguyen Thi, Linh Ha My, Hung Nguyen Viet, Huyen Nguyen Thi Minh, and Phuong Le Hong. 2013. Building a treebank for Vietnamese dependency parsing. In *Proceedings of RIVF*, pages 147–151.

Thanh Vu, Dat Quoc Nguyen, Dai Quoc Nguyen, Mark Dras, and Mark Johnson. 2018. VnCoreNLP: A Vietnamese Natural Language Processing Toolkit. In *Proceedings of NAACL: Demonstrations*, pages 56–60.

Daniel Zeman et al. 2017. CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies. In *Proceedings of the CoNLL 2017 Shared Task*, pages 1–19.

Daniel Zeman et al. 2018. CoNLL 2018 shared task: Multilingual parsing from raw text to universal dependencies. In *Proceedings of the CoNLL 2018 Shared Task*, pages 1–21.

Meishan Zhang, Yue Zhang, Wanxiang Che, and Ting Liu. 2014. Character-Level Chinese Dependency Parsing. In *Proceedings of ACL*, pages 1326–1336.

Xingxing Zhang, Jianpeng Cheng, and Mirella Lapata. 2017. Dependency Parsing as Head Selection. In *Proceedings of EACL*, pages 665–676.

Yuan Zhang, Chengtao Li, Regina Barzilay, and Kareem Darwish. 2015. Randomized Greedy Inference for Joint Segmentation, POS Tagging and Dependency Parsing. In *Proceedings of NAACL-HLT*, pages 42–52.

Yuan Zhang and David Weiss. 2016. Stack-propagation: Improved representation learning for syntax. In *Proceedings of ACL*, pages 1557–1566.

Appendix

Implementation details: When training, each task component is fed with the corresponding task-associated sentences. The dependency parsing training set is smallest in size (consisting of 8,977 sentences), thus for each training epoch, we sample the same number of sentences from the word segmentation and POS tagging training sets. We train our model with a fixed random seed and without mini-batches. Dropout (Srivastava et al., 2014) is applied with a 67% keep probability to the inputs of BiLSTMs and FFNNs. Following Kiperwasser and Goldberg (2016), we also use *word dropout* to learn embeddings for unknown syllables/words: we replace each syllable/word token s/w appearing $\#(s/w)$ times with a "unk" symbol with probability $\mathsf{p}_{unk}(s/w) = \frac{0.25}{0.25+\#(s/w)}$.

We initialize syllable and word embeddings with 100-dimensional pre-trained Word2Vec vectors as used in Nguyen et al. (2017), while the initial word-boundary and POS tag embeddings are randomly initialized. All these embeddings are then updated when training. The sizes of the output layers of FFNN_{WS}, FFNN_{POS} and $\text{FFNN}_{\text{LABEL}}$ are the number of BIO word-boundary tags (i.e. 3), the number of POS tags and the number of dependency relation types, respectively. We perform a minimal grid search of hyper-parameters, resulting in the size of the initial word-boundary tag embeddings at 25, the POS tag embedding size of 100, the size of the output layers of remaining FFNNs at 100, the number of BiLSTM layers at 2 and the size of LSTM hidden states in each layer at 128.

Additional results: Table 3 presents POS tagging and parsing accuracies w.r.t. gold word segmentation. In this case, for our jointWPD, we feed the POS tagging and parsing components with gold word-segmented sentences when decoding.

	Model	WSeg	PTag	LAS	UAS
Gold segment.	Our jointWPD	100.0	95.97	73.90	80.12
	VnCoreNLP	100.0	95.88	71.38^{**}	77.35^{**}
	jPTDP-v2	100.0	95.70^{*}	73.12^{**}	79.63^{*}
	Biaffine	100.0	95.88	74.99^{**}	81.19^{**}

Table 3: POS tagging, LAS and UAS accuracy scores on the test sets w.r.t. gold word-segmented sentences. These scores are computed on all tokens (including punctuation). Recall that the LAS and UAS accuracies are computed on the VnDT v1.1 test set w.r.t. the automatically predicted POS tags.