Sequence-Labeling RoBERTa Model for Dependency-Parsing in Classical Chinese and Its Application to Vietnamese and Thai

Koichi Yasuoka
Institute for Research in Humanities
Kyoto University
Kyoto 606-8265 JAPAN
yasuoka@kanji.zinbun.kyoto-u.ac.jp

***Abstract*—The author and his colleagues have been developing classical Chinese treebank using Universal Dependencies. We also developed RoBERTa-Classical-Chinese model pre-trained with classical Chinese texts of 1.7 billion characters.**

In this paper we describe how to finetune sequence-labeling RoBERTa model for dependency-parsing in classical Chinese. We introduce "goeswith"-labeled edges into the directed acyclic graphs of Universal Dependencies in order to resolve the mismatch between the token length of RoBERTa-Classical-Chinese and the word length in classical Chinese. We utilize [MASK]-token of RoBERTa model to handle outgoing edges and to produce the adjacency-matrices for the graphs of Universal Dependencies. Our RoBERTa-UDgoeswith model outperforms other dependency-parsers in classical Chinese on LAS / MLAS / BLEX benchmark scores.

Then we apply our methods to other isolating languages. For Vietnamese we introduce "goeswith"-labeled edges to separate words into space-separated syllables, and finetune RoBERTa and PhoBERT models. For Thai we try three kinds of tokenizers, character-wise tokenizer, quasi-syllable tokenizer, and SentencePiece, to produce RoBERTa models.

***Index Terms*—dependency-parsing, part-of-speech tagging, sequence-labeling, Universal Dependencies, pre-trained language model**

I. Introduction

On May 15, 2019, the author and his colleagues released the first version of UD_Classical_Chinese-Kyoto treebank (11,176 sentences, 55,026 words, 56,768 characters) as a part of Universal Dependencies (UD) 2.4 [1]. The treebank consisted of the Four Books (孟子, 論語, 大學, and 中庸; taken from Kanseki Repository [2]) with Part-Of-Speech (POS) tags and manually-annotated dependency relations. We continue developing the treebank and in UD 2.11 (dated November 15, 2022) we have included 孟子, 論語, 禮記, 十八史略, 楚辭, 唐詩三百首, 摩訶般若波羅蜜大明呪經, 金剛般若波羅蜜經, and 佛說阿彌陀經 (total 63,079 sentences, 310,594 words, 324,415 characters). The treebank is now utilized by several dependency-parsers, such as Stanza [3], Trankit [4], and UDPipe 2 [5].

We also developed RoBERTa-Classical-Chinese [6], which was derived from GuwenBERT [7], as a pre-trained model for classical Chinese (1.7 billion characters). And we have developed a sequence-labeling RoBERTa model for dependency-parsing in classical Chinese [8], finetuning RoBERTa-Classical-Chinese with UD_Classical_Chinese-Kyoto treebank. Now we apply the method to other isolating languages, such as Vietnamese, Thai, and modern Chinese. In this paper the author describes about the sequence-labeling RoBERTa models for dependency-parsing in classical Chinese, Vietnamese, Thai, and modern Chinese.

II. Brief Description of Universal Dependencies

UD [9] represents natural language texts as directed acyclic graphs of words. Every word has a single incoming edge, which is connected from HEAD of the word and is labeled by DEPREL (Table I). Each graph is stored in CoNLL-U format, tab-separated ten-column lines (UTF-8):

1. ID: Word index, integer starting at 1 for each new sentence.
2. FORM: Word form or punctuations symbol.
3. LEMMA: Lemma or stem of word form.
4. UPOS: Universal POS tag (ADJ ADP ADV AUX CCONJ DET INTJ NOUN NUM PART PRON PROPN PUNCT SCONJ SYM VERB X).
5. XPOS: Language-specific POS tag; underscore if not available.
6. FEATS: List of morphological features from the universal feature inventory or from a defined language-specific extension.
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 iff HEAD = 0) or a defined language-specific subtype of one.
9. DEPS: Enhanced dependency graph in the form of a list of head-deprel pairs.
10. MISC: Any other annotation.

Fig. 1 and 2 show example sentences in CoNLL-U format and directed graph representation by "deplacy" [10], respectively.

```
# text = 君子周而不比
1	君子	君子	NOUN	_	_	2	nsubj	_	SpaceAfter=No
2	周	周	VERB	_	_	0	root	_	SpaceAfter=No
3	而	而	CCONJ	_	_	5	cc	_	SpaceAfter=No
4	不	不	ADV	_	Polarity=Neg	5	advmod	_	SpaceAfter=No
5	比	比	VERB	_	_	2	conj	_	SpaceAfter=No

# text = Quân tử chu toàn mà không so sánh
1	Quân tử	quân tử	NOUN	_	_	2	nsubj	_	_
2	chu toàn	chu toàn	ADJ	_	_	0	root	_	_
3	mà	mà	SCONJ	_	_	5	cc	_	_
4	không	không	ADV	_	Polarity=Neg	5	advmod	_	_
5	so sánh	so sánh	VERB	_	_	2	conj	_	SpaceAfter=No

# text = วิญญูชนสมัครสมานแต่ไม่สมคบคิด
1	วิญญูชน	วิญญูชน	NOUN	_	_	2	nsubj	_	SpaceAfter=No
2	สมัคร	สมัคร	VERB	_	_	0	root	_	SpaceAfter=No
3	สมาน	สมาน	ADV	_	_	2	advmod	_	SpaceAfter=No
4	แต่	แต่	CCONJ	_	_	6	cc	_	SpaceAfter=No
5	ไม่	ไม่	PART	_	Polarity=Neg	6	advmod	_	SpaceAfter=No
6	สมคบ	สมคบ	VERB	_	_	2	conj	_	SpaceAfter=No
7	คิด	คิด	VERB	_	_	6	advcl	_	SpaceAfter=No
```

Fig. 1. CoNLL-U in classical Chinese, Vietnamese, and Thai

TABLE I
UNIVERSAL DEPENDENCY RELATIONS (DEPREL)

	Nominals	**Clauses**	**Modifier Words**	**Function Words**
Core arguments	nsubj obj iobj	csubj ccomp xcomp		
Non-core dependents	obl vocative expl dislocated	advcl	advmod discourse	aux cop mark
Nominal dependents	nmod appos nummod	acl	amod	det clf case
Coordination	**MWE**	**Loose**	**Special**	**Other**
conj cc	fixed flat compound	list parataxis	orphan goeswith reparandum	punct root dep

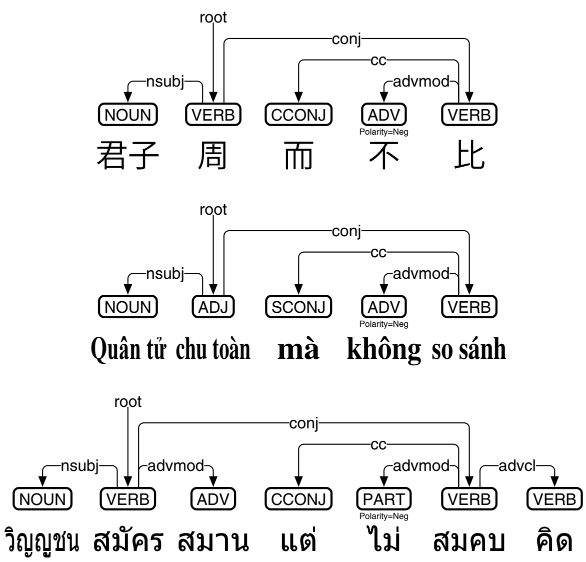

Fig. 2. Directed graph representation of Fig. 1

III. SEQUENCE-LABELING ROBERTA MODELS FOR DEPENDENCY-PARSING

We use the triplet <UPOS FEATS DEPREL> for the labels to finetune sequence-labeling RoBERTa models. We additionally use the special triplet <X _ goeswith> to indicate the second and following tokens of a word, when RoBERTa separates the word into two or more tokens. We also use "Null" label for special-tokens (such as [CLS]-token to start a sentence, [SEP]-token to end a sentence, and [PAD]-token for padding), and for the detection of outgoing edges with [MASK]-token as mentioned below.

A. Classical Chinese

Fig. 3 shows token-label pairs of "君子周而不比" for sequence-labeling, where 子 is labeled by the special

[CLS]	君	子	周	而	不	比	[SEP]	[PAD]
Null	<NOUN _ nsubj>	<X _ goeswith>	<VERB _ root>	<CCONJ _ cc>	<ADV Polarity=Neg advmod>	<VERB _ conj>	Null	Null

Fig. 3. Token-label pairs of "君子周而不比" for sequence-labeling

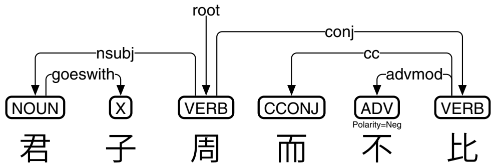

Fig. 4. <X _ goeswith> extended graph of “君子周而不比”

[CLS]	[MASK]	子	周	而	不	比	[SEP]	君
Null	Null	<X _ goeswith>	Null	Null	Null	Null	Null	Null
[CLS]	君	[MASK]	周	而	不	比	[SEP]	子
Null	Null	Null	Null	Null	Null	Null	Null	Null
[CLS]	君	子	[MASK]	而	不	比	[SEP]	周
Null	<NOUN _ nsubj>	Null	<VERB _ root>	Null	Null	<VERB _ conj>	Null	Null
[CLS]	君	子	周	[MASK]	不	比	[SEP]	而
Null	Null	Null	Null	Null	Null	Null	Null	Null
[CLS]	君	子	周	而	[MASK]	比	[SEP]	不
Null	Null	Null	Null	Null	Null	Null	Null	Null
[CLS]	君	子	周	而	不	[MASK]	[SEP]	比
Null	Null	Null	Null	<CCONJ _ cc>	<ADV|Polarity=Neg|advmod>	Null	Null	Null

Fig. 5. Sequences for dependency-parsing of “君子周而不比”

triplet <X _ goeswith> since the character-wise tokenizer of RoBERTa-Classical-Chinese separates the word 君子 into two tokens.

Here we extend the sequence-labeling to handle outgoing edges of a directed acyclic graph of tokens, borrowing the idea of [11]. In fact, our extension is a hybrid[1] of two state-of-the-art algorithms in POS-tagging and dependency-parsing: jPTDP [12] and Biaffine [13].

For example in the graph of Fig. 4, 周 has one root and two outgoing edges, nsubj to 君 and conj to 比. We permutate the sequence shown in Fig. 3, substituting [MASK]-token for 周 and moving 周 after [SEP]-token, then we leave three triplets: <NOUN _ nsubj> <VERB _ root> and <VERB _ conj>, while replacing other triplets into “Null” to produce the third row of Fig. 5. We apply the same way to other five tokens in Fig. 4, then we obtain six sequences as shown in Fig. 5, which involve (between [CLS] and [SEP]) the 6×6 adjacency-matrix of the graph in Fig. 4.

Following the strategies as mentioned above (both in Fig. 3 and 5), the author finetuned RoBERTa-Classical-Chinese (base model) with UD_Classical_Chinese-Kyoto treebank into RoBERTa-UDgoeswith, a sequence-labeling RoBERTa model for dependency-parsing in classical Chinese, and implemented the dependency-parser with RoBERTa-UDgoeswith as a pipeline of Transformers [18].

[1] The author and his colleagues had investigated which dependency-parsing algorithm was suitable to analyze classical Chinese, such as arc-planar [14] used in UDPipe [15], arc-swap [16] used in spaCy [17], Biaffine [13] used in Stanza [3], joint POS-Tagging and Dependency-Parsing (jPTDP) [12], and so on [1], [6], [8]. Then we decided to use Biaffine modified with jPTDP, fitting them for RoBERTa-Classical-Chinese by “goeswith”-labeled edges.

TABLE II
LAS / MLAS / BLEX IN CLASSICAL CHINESE (UD 2.11)

	lzh_kyoto-ud-dev.conllu	lzh_kyoto-ud-test.conllu
RoBERTa-UDgoeswith	**80.87 / 77.87 / 79.33**	**81.43 / 78.09 / 79.59**
Stanza 1.4.2	68.32 / 64.70 / 67.44	72.10 / 67.97 / 70.86
Trankit 1.1.1	58.28 / 52.88 / 56.76	68.59 / 63.37 / 67.01
UDPipe 2	68.28 / 64.35 / 67.36	71.28 / 66.93 / 69.83

```
!pip install transformers stanza trankit deplacy
import os,sys,subprocess,stanza,trankit
from transformers import pipeline
from deplacy import to_conllu
url="https://github.com/UniversalDependencies"
d="UD_Classical_Chinese-Kyoto"
os.system("test -d {} || git clone --depth=1 {}/{}".format(d,url,d))
url="https://universaldependencies.org/conll18"
c="conll18_ud_eval.py"
os.system("test -f {} || curl -LO {}/{}".format(c,url,c))
class UDPipe2WebAPI(object):
  def __init__(self,lang):
    self.url="https://lindat.mff.cuni.cz/services/udpipe/api/process"
    self.params={"model":lang,"tokenizer":"","tagger":"","parser":""}
  def __call__(self,text):
    import urllib.parse,urllib.request,json
    self.params["data"]=urllib.parse.quote(text)
    u=self.url+"?"+"&".join(k+"="+v for k,v in self.params.items())
    with urllib.request.urlopen(u) as r:
      return json.loads(r.read())["result"]
for p in [lambda x:pipeline(task="universal-dependencies",model=x[0],
  trust_remote_code=True,aggregation_strategy="simple"),
  lambda x:stanza.Pipeline(x[2]),lambda x:trankit.Pipeline(x[1]),
  lambda x:UDPipe2WebAPI(x[2])]:
  nlp=p(["KoichiYasuoka/roberta-classical-chinese-base-ud-goeswith",
    "classical-chinese","lzh"])
  for f in ["lzh_kyoto-ud-dev.conllu","lzh_kyoto-ud-test.conllu"]:
    with open(os.path.join(d,f),"r",encoding="utf-8") as r:
      s=r.read()
    with open("result.conllu","w",encoding="utf-8") as w:
      for t in s.split("\n"):
        if t.startswith("# text = "):
          w.write(to_conllu(nlp(t[9:])))
    print("*** "+f+" "+str(type(nlp)),subprocess.check_output(
      [sys.executable,c,os.path.join(d,f),"result.conllu"],
      encoding="utf-8"),sep="\n")
```

Fig. 6. Benchmark script on Google Colaboratory

Table II shows LAS (Labeled Attachment Score) / MLAS (Morphology-aware Labeled Attachment Score) / BLEX (Bi-LEXical dependency score) [19] of RoBERTa-UDgoeswith, using the benchmark script (Fig. 6) on `dev` and `test` of UD_Classical_Chinese-Kyoto treebank in UD 2.11, comparing with Stanza, Trankit, and UDPipe 2 WebAPI.

B. Vietnamese

Vietnamese words often consist of two or more syllables that are separated by spaces. It means that such words include spaces, as “Quân tử” “chu toàn” and “so sánh” in Fig. 1 and 2. The tokens of RoBERTa-Vietnamese are mainly syllables, thus we use the special triplet <X _ goeswith> to handle such

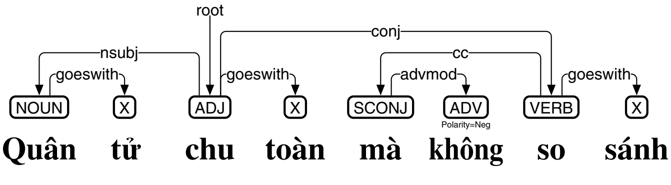

Fig. 7. <X _ goeswith> extended graph in Vietnamese

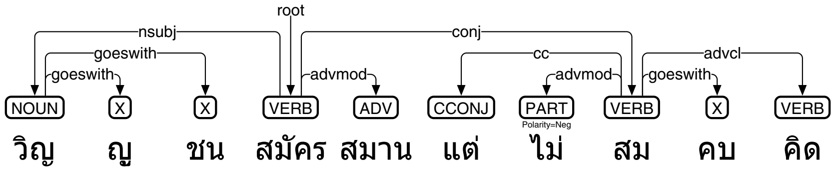

Fig. 8. <X _ goeswith> extended graph in Thai with quasi-syllable tokenizer

TABLE III
LAS / MLAS / BLEX IN VIETNAMESE (UD 2.11)

	vi_vtb-ud-dev.conllu	vi_vtb-ud-test.conllu
RoBERTa-UDgoeswith	69.17 / 60.28 / 59.99	70.48 / 61.14 / 61.27
PhoBERT-UDgoeswith	**70.91 / 63.24 / 61.86**	**72.75 / 64.41 / 63.99**
Stanza 1.4.2	41.28 / 19.59 / 34.15	41.46 / 18.83 / 35.82
Trankit 1.1.1	54.21 / 26.77 / 45.72	55.14 / 25.84 / 48.57
UDPipe 2	37.21 / 16.96 / 31.05	36.86 / 15.42 / 31.81

words (Fig. 7). On the other hand, the tokens of PhoBERT [20], an extended RoBERTa model for Vietnamese, are partly syllables and partly words, thus we use both Vietnamese graphs as in Fig. 7 and 2, with and without "goeswith"-labeled edges.

Table III shows LAS / MLAS / BLEX of RoBERTa-UDgoeswith and PhoBERT-UDgoeswith (base models) on `dev` and `test` of UD_Vietnamese-VTB treebank in UD 2.11, comparing with Stanza, Trankit, and UDPipe 2 WebAPI.

C. Thai

For RoBERTa-Thai we have prepared three kinds of tokenizers: SentencePiece [21] unigram tokenizer (spm) whose maximum piece length is four, quasi-syllable tokenizer [22] borrowed from WangchanBERTa [23], and character-wise tokenizer. For example, spm separates **วิญญูชน** into six tokens, quasi-syllable into three, and character-wise into seven. Thus **วิญญูชน** requires five "goeswith"-labeled edges with spm, two with quasi-syllable (as shown in Fig. 8), six with character-wise.

Table IV shows LAS / MLAS / BLEX of three RoBERTa-UDgoeswith models on UD_Thai-Corpora treebank of spaCy-Thai [10], comparing with spaCy-Thai since other dependency-parsers do not support Thai.

TABLE IV
LAS / MLAS / BLEX COMPARISON WITH SPACY-THAI

	th_lst-ud-dev.conllu	th_lst-ud-test.conllu
RoBERTa-UDgoeswith spm	**77.53** / **71.49** / 75.50	**65.66** / 56.11 / **63.50**
quasi-syllable	77.04 / 70.47 / **75.52**	64.87 / **56.71** / 63.11
character-wise	69.77 / 61.57 / 66.76	56.79 / 47.54 / 53.87
spaCy-Thai 0.7.3	35.07 / 28.84 / 36.11	36.19 / 23.24 / 34.15

D. Modern Chinese

We made tentative trial to produce RoBERTa model in modern Chinese, pre-trained with mixed texts in simplified and traditional Chinese (0.9 billion characters) on character-wise tokenizer. To finetune RoBERTa-UDgoeswith model we followed the same strategies as used in classical Chinese. However, as shown in Tables V and VI on UD_Chinese-GSDSimp and UD_Chinese-GSD treebanks in modern Chinese, our RoBERTa-UDgoeswith model could not slightly reach to Trankit whose tokenizer is well-tuned[2] to tokenize simplified and traditional Chinese.

TABLE V
LAS / MLAS / BLEX IN SIMPLIFIED CHINESE (UD 2.11)

	zh_gsdsimp-ud-dev.conllu	zh_gsdsimp-ud-test.conllu
RoBERTa-UDgoeswith	74.13 / 72.41 / 76.66	**75.45** / **73.49** / 77.58
Stanza 1.4.2	65.82 / 63.29 / 67.11	68.08 / 65.05 / 69.00
Trankit 1.1.1	**74.47 / 72.94 / 77.87**	74.86 / 72.99 / **78.06**
UDPipe 2	64.87 / 62.03 / 65.74	66.63 / 63.68 / 67.31

TABLE VI
LAS / MLAS / BLEX IN TRADITIONAL CHINESE (UD 2.11)

	zh_gsd-ud-dev.conllu	zh_gsd-ud-test.conllu
RoBERTa-UDgoeswith	74.07 / 72.21 / 76.52	75.32 / 73.38 / 77.48
Stanza 1.4.2	64.48 / 61.60 / 65.52	67.04 / 63.62 / 67.81
Trankit 1.1.1	**74.55 / 73.05 / 77.98**	**75.71 / 74.40 / 79.05**
UDPipe 2	64.44 / 61.55 / 65.09	66.63 / 63.69 / 67.38

IV. CONCLUSION

We have developed sequence-labeling RoBERTa model for dependency-parsing in classical Chinese. Our model outperforms other dependency-parsers, such as Stanza, Trankit, and UDPipe 2, in classical Chinese. We have developed RoBERTa-UDgoeswith and PhoBERT-UDgoeswith models in Vietnamese, and PhoBERT-UDgoeswith outperforms other dependency-parsers. We have developed three RoBERTa-UDgoeswith models in Thai, and got that SentencePiece (spm) and quasi-syllable models are better than character-wise. We need to resume developing in modern Chinese, using tokenizers other than character-wise.

We have published these RoBERTa models on WWW at `https://huggingface.co/KoichiYasuoka/` and we have a plan to include these RoBERTa models in our "esupar" multilingual dependency-parser package.

[2] In our humble opinion, the tokenizer of Trankit is well-tuned specially to tokenize foreign names, which often consist of four or more characters in modern Chinese.

REFERENCES

[1] K. Yasuoka, "Universal Dependencies Treebank of the Four Books in Classical Chinese," DADH2019: 10th International Conference of Digital Archives and Digital Humanities, pp. 20–28, December 2019.

[2] C. Wittern, "Special Issue: Kanseki Repository," CIEAS Research Report 2015, Kyoto University, March 2016.

[3] P. Qi, Y. Zhang, Y. Zhang, J. Bolton, and C. D. Manning, "Stanza: A Python Natural Language Processing Toolkit for Many Human Languages," 58th Annual Meeting of the Association for Computational Linguistics, Proceedings of the System Demonstration, pp. 101–108, July 2020.

[4] M. V. Nguyen, V. D. Lai, A. P. B. Veyseh, and T. H. Nguyen, "Trankit: A Light-Weight Transformer-based Toolkit for Multilingual Natural Language Processing," EACL 2021: 16th Conference of the European Chapter of the Association for Computational Linguistics, System Demonstrations, pp. 80–90, April 2021.

[5] M. Straka: "UDPipe 2.0 Prototype at CoNLL 2018 UD Shared Task," Proceedings of the CoNLL 2018 Shared Task, pp. 197–207, October 2018.

[6] K. Yasuoka, C. Wittern, T. Morioka, T. Ikeda, N. Yamazaki, Y. Nikaido, S. Suzuki, S. Moro, and K. Fujita, "Designing Universal Dependencies for Classical Chinese and Its Application," Journal of Information Processing, vol. 63, no. 2, pp. 355–363, February 2022.

[7] T. Yan and Z. Chi, "基于继续训练的古汉语预训练语言模型," 19th China National Conference on Computational Linguistics, "古联杯"古籍文献命名实体识别, October 2020.

[8] K. Yasuoka and M. Yasuoka, "古典中国語の形態素解析と係り受け解析," 근역한문학회 2022년 추계 기획학술대회: 디지털과 한문 고전 연구, pp. 148–160, November 2022.

[9] M.-C. de Marneffe, C. D. Manning, J. Nivre, and D. Zeman, "Universal Dependencies," Computational Linguistics, vol. 47, no. 2, pp. 255–308, June 2021.

[10] K. Yasuoka, "deplacy: a CUI-based tree visualizer for Universal Dependencies," IPSJ Symposium Series, vol. 2020, no. 1, pp. 95–100, December 2020.

[11] L. Màrquez, P. Comas, J. Giménez, and N. Català, "Semantic Role Labeling as Sequential Tagging," Proceedings of the 9th Conference on Computational Natural Language Learning, pp. 193–196, June 2005.

[12] D. Q. Nguyen and K. Verspoor, "An improved neural network model for joint POS tagging and dependency parsing," Proceedings of the CoNLL 2018 Shared Task, pp. 81–91, October 2018.

[13] T. Dozat and C. D. Manning, "Deep Biaffine Attention for Neural Dependency Parsing," 5th International Conference on Learning Representations, C25, April 2017.

[14] C. Gómez-Rodríguez and J. Nivre, "A Transition-Based Parser for 2-Planar Dependency Structures," 48th Annual Meeting of the Association for Computational Linguistics, pp. 1492–1501, July 2010.

[15] M. Straka and J. Straková, "Tokenizing, POS Tagging, Lemmatizing and Parsing UD 2.0 with UDPipe," Proceedings of the CoNLL 2017 Shared Task, pp. 88–99, August 2017.

[16] J. Nivre, "Non-Projective Dependency Parsing in Expected Linear Time," Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP, pp. 351–359, August 2009.

[17] M. Honnibal and M. Johnson, "An Improved Non-monotonic Transition System for Dependency Parsing," EMNLP 2015: Conference on Empirical Methods in Natural Language Processing, pp. 1373–1378, September 2015.

[18] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. Le Scao, S. Gugger, M. Drame, Q. Lhoest, and A. M. Rush, "Transformers: State-of-the-Art Natural Language Processing," EMNLP 2020: Conference on Empirical Methods in Natural Language Processing, System Demonstrations, pp. 38–45, October 2020.

[19] D. Zeman, J. Hajič, M. Popel, M. Potthast, M. Straka, F. Ginter, J. Nivre, and S. Petrov, "CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies," Proceedings of the CoNLL 2018 Shared Task, pp. 1–21, October 2018.

[20] D. Q. Nguyen and A. T. Nguyen, "PhoBERT: Pre-trained language models for Vietnamese," Findings of the Association for Computational Linguistics: EMNLP 2020, pp. 1037–1042, November 2020.

[21] T. Kudo and J. Richardson, "SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing," EMNLP 2018: Conference on Empirical Methods in Natural Language Processing, System Demonstrations, pp. 66–71, November 2018.

[22] P. Chormai, P. Prasertsom, J. Cheevaprawatdomrong, and A. T. Rutherford, "Syllable-based Neural Thai Word Segmentation," Proceedings of the 28th International Conference on Computational Linguistics, pp. 4619–4637, December 2020.

[23] L. Lowphansirikul, C. Polpanumas, N. Jantrakulchai, and S. Nutanong, "WangchanBERTa: Pretraining transformer-based Thai Language Models," arXiv:2101.09635, March 2021.