# Graph-based Dependency Parser Building for Myanmar Language

**Zar Zar Hlaing**, Ye Kyaw Thu, Thepchai Supnithi, and Ponrudee Netisopakul

**iSAI-NLP-AIoT 2022** **November 5 - 7**


NECTEC
a member of NSTDA

# CONTENTS

# 1. Introduction

- Dependency parsing analyzes sentence structure through sentence parsing
- It has been proved to be one of the most useful preprocessing steps in NLP tasks
- Dependency structure is more simple than a phrase to present syntatic and semantic information
- An input sentence is parsed using dependency parser to create a rooted dependency tree
- Nodes represent words and the edges are the syntactic relationships

# 1. Introduction (Cont.)

- Dependency parsing models can be divided into graph-based and transition-based models
- Performance of some NLP tasks depend on more accurate dependency parsers
- There are several dependency parsers for English and other resource-rich languages
- For the Myanmar language, there is no publicly available dependency parser

# 1. Introduction (Cont.)

- Large amount of Myanmar UD corpus is required
- Graph-based and transition-based models for dependency parsing
- Selected *joint POS tagging and dependency parsing* model (jPTDP) and *universal graph-based parsing* model (UniParse)
- Simple, flexible and easy-to-customized graph-based neural parsing models

# 2. Objective

**Objectives of this study:**

- To manually extend the existing small amount of Myanmar UD corpus
- To build the dependency parser using the extended UD corpus
- To compare the evaluation scores between dependency parsing models
- To apply the dependency information to FNMT system for low-resource language Myanmar in future
- To publish the extended Myanmar UD corpus and our built dependency parser

# 3. Related Work

- *Hnin Thu Zar Aye et al.* (2018) annotated a general domain corpus
  - to build Treebank for unsupervised dependency parsing of the Myanmar Language
  - UDPipe (Straka et al., 2016) and the shared Japanese dependency model were used
  - first constructed the Myanmar UD corpus
  - we extended the original Myanmar UD corpus and built the dependency parser.
- *jPTDP* model was proposed by *Nguyen* and *Verspoor* (2018)
  - extended a tagging component based on BiLSTM to well-known BIST graph-based dependency parser (Kiperwasser and Goldberg, 2016)
  - proved that their proposed model outperformed the baseline UDPipe
- Graph-based dependency parsers are conceptually simple
  - a severe lack of extensible and modular implementations for long-term dependency parsers
  - Varab et al. (2014) implemented a flexible and highly expressive framework for graph-based dependency parsing architecture, namely, *UniParse*
  - *UniParse* aimed to provide strong baseline for future research on graph-based dependency parsers

# 4. Myanmar Dependency Structure

- Each phrase is dependent on primary sentence root
- Source word depends on another word in dependency structure of a phrase
- When clauses are present, their roots are dependent on the root of the phrase
- A sentence may include zero or more clauses
- Dependency structures cover the fundamental structures (noun, verb, adjective, adverb, and conjunction)

# 4. Myanmar Dependency Structure (Cont.)

## Dependency Relations

- Dependency relations that are most primarily used in Myanmar dependency structure

1. *root* (root)
2. *acl* (clausal modifier of noun)
3. *amod* (adjectival modifier)
4. *advmod* (adverbial modifier)
5. *case* (case marking)
6. *mark* (marker)
7. *compound* (compound)
8. *obl* (oblique nominal)
9. *obj* (object)
10. *punct* (punctuation)

# 4. Myanmar Dependency Structure (Cont.)

## 4.1. The *root* dependency relation

- points to the root of the sentence
- every dependency tree should have only one node that has root
- if the main predicate is not present and there are multiple orphaned dependents, one of these is promoted to *root*
- sample of *root* dependency relation is illustrated in Figure 1

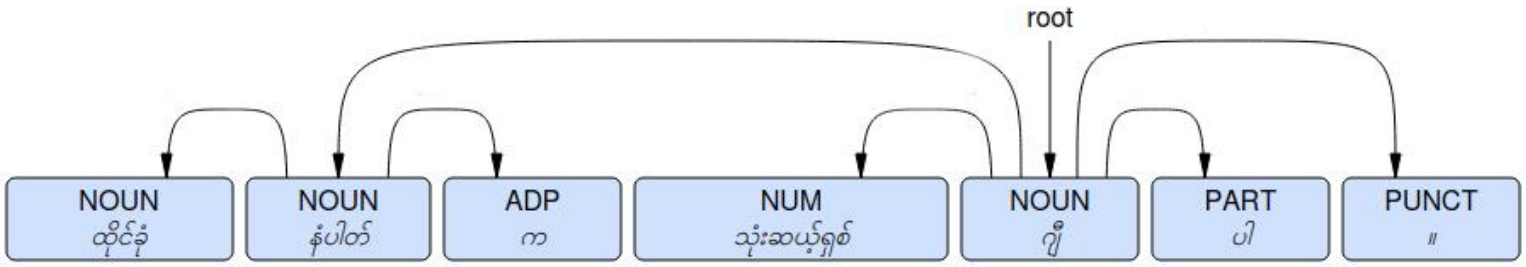


Fig. 1. The *root* dependency relation for Myanmar sentence "ထိုင်ခုံ နံပါတ် က သုံးဆယ့်ရှစ် ဂျီ ပါ ။" ("The seat number is 38 G." in English).

# 4. Myanmar Dependency Structure (Cont.)

## 4.2. The *acl* dependency relation

- stands for finite and non-finite clauses that modify a nominal
- head of the *acl* relation is the noun that is modified, and the dependent is the head of the clause that modifies the noun
- Figure 2 shows *acl* dependency relation

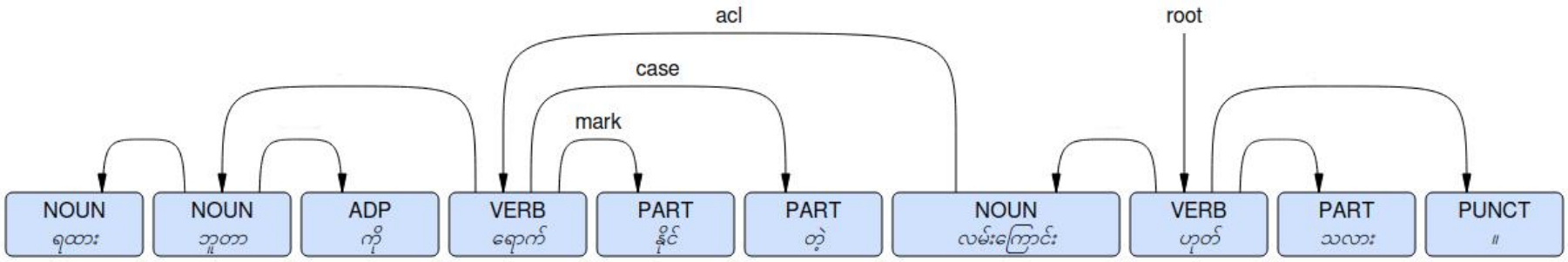


Fig. 2. The *acl* dependency relation for Myanmar sentence "ရထား ဘူတာ ကို ရောက် နိုင် တဲ့ လမ်းကြောင်း ဟုတ် သလား ။" ("Is there a way to get to the train station?" in English).

# 4. Myanmar Dependency Structure (Cont.)

## 4.3. The *amod* dependency relation

- any adjectival phrase that serves to modify a noun is said to be its adjectival modifier (*amod*)
- *amod* dependents are allowed to have their own modifiers, but they shouldn't be clauses
- *acl* should be used if it is a clause
- sample of *amod* dependency relation is shown in Figure 3

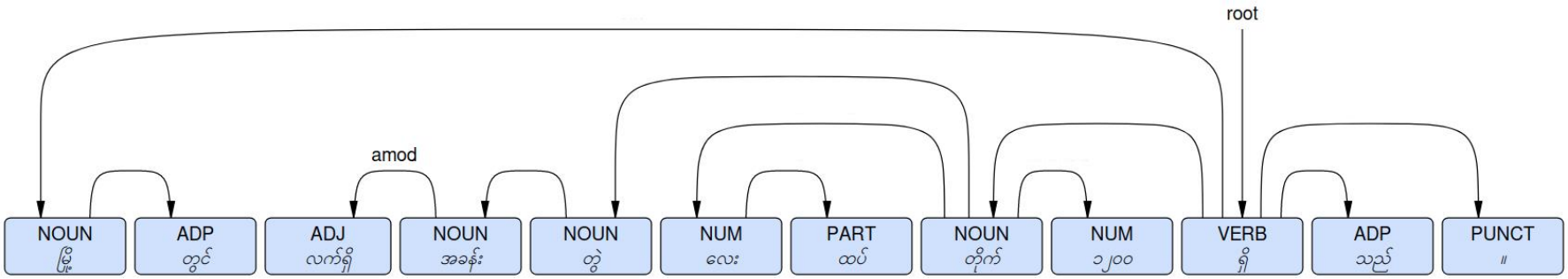


Fig. 3. The *amod* dependency relation for Myanmar sentence "မြို့ တွင် လက်ရှိ အခန်း တွဲ လေး ထပ် တိုက် ၁၂၀၀ ရှိ သည် ။" ("There are currently 1,200 four-story apartments in the city." in English).

# 5. Methodologies

## 5.1. Joint POS Tagging and Graph-based Dependency Parsing Model (jPTDP)

- it can be viewed as a two-component mixture
- tagging component uses a BiLSTM to learn “latent” feature vectors
- feeds these feature vectors into a multilayer perceptron
- parsing component uses a different BiLSTM
- two different MLPs: one to decode dependency arcs and the other to label predicted dependency arcs

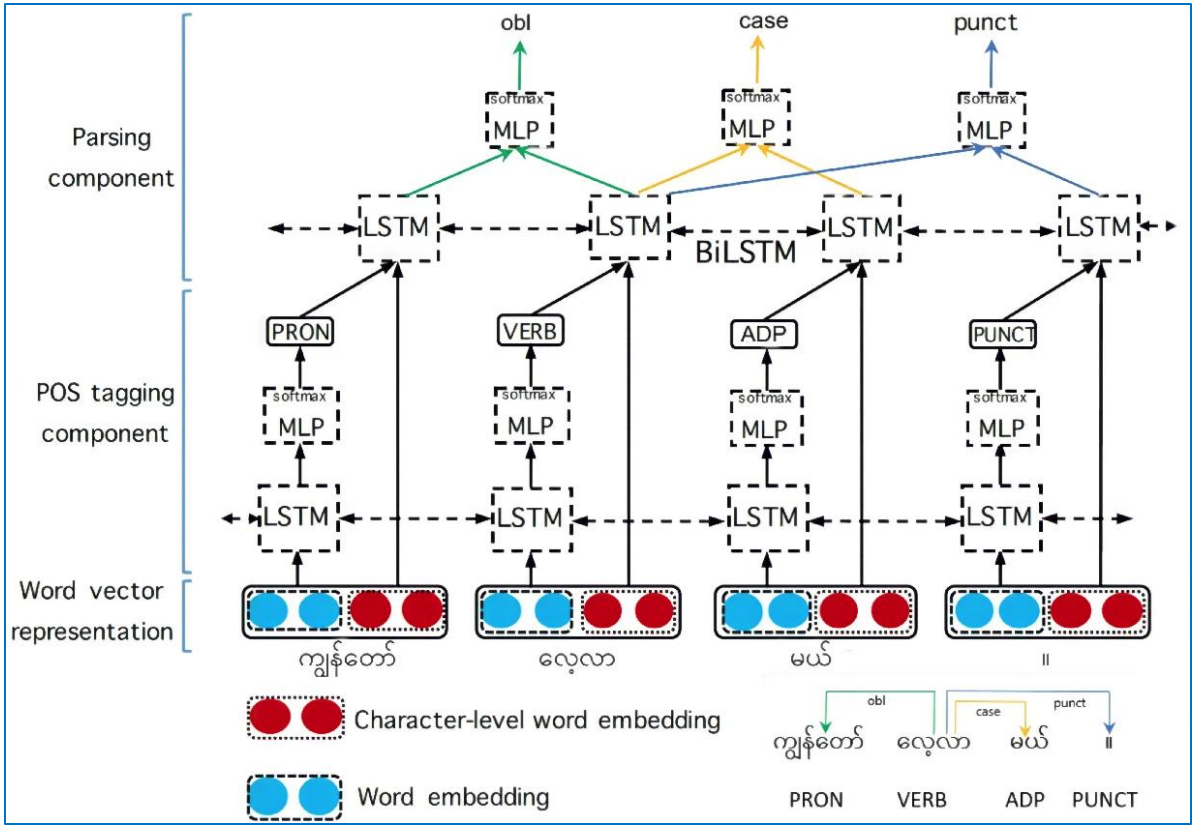


Fig. 4. Architecture of Joint POS Tagging and Graph-based Dependency Parsing Model

# 5. Methodologies (Cont.)

## 5.2. Universal Graph-based Parsing Model (*UniParse*)

- *UniParse* was also used
- is made up of three components including an encoder $\Gamma$, a set of parameters $\lambda$, and a decoder *h*
- possible dependency relations between all words of a sentence S are modeled as a complete directed graph $G_S$
- arc in $G_S$ referred to a factor that $\Gamma$ associates with a *d*-dimensional feature vector, and its encoding
- set of parameters $\lambda$ is then used to generate the scores
- a decoder *h* outputs a well-formed dependency tree

# 6. Experimental Setup

## 6.1. Myanmar UD Corpus Extension

- data from original myPOS UD corpus were taken from *economics*, *history*, *news*, and *politics* domain areas of Wikipidea
- original corpus is small and may not cover some other domain areas
- we extended the original UD corpus in this study
- built Myanmar dependency parser using jPTDP model with the original UD corpus
- 20,052 Myanmar sentences from our developing parallel corpora and 12,144 Myanmar sentences from ASEAN MT corpus were parsed

# 6. Experimental Setup (Cont.)

## 6.1. Myanmar UD Corpus Extension

- manually checked and corrected the parsed data using the CoNLL-U viewer (https://urd2.let.rug.nl/~kleiweg/conllu/)
- revised data are then combined with the original myPOS version 1.0 UD corpus
- data statistics of extended UD corpus is shown in Table 1

**Table 1.** Data Statistics of myPOS (version 3.0) UD Corpus

| Unit | myPOS (ver. 1.0) | Ext-1: my-zh | Ext-2: my-ko | Ext-3: ASEAN-MT my | myPOS (ver. 3.0) |
|---|---|---|---|---|---|
| Sentences | 11,000 | 10,000 | 10,052 | 12,144 | 43,196 |
| Word Tokens | 239,598 | 103,909 | 106,864 | 114,134 | 564,505 |
| Average Words/Sentence | 21.78 | 10.17 | 10.64 | 9.40 | 13.07 |

# 6. Experimental Setup (Cont.)

## 6.2. Experimental Settings

- used default parameter settings
- for the comparison of dependency parsing models, two training data sets are used
- one for the baseline model is taken from the original myPOS UD corpus (10,000 sentences)
- the other is also taken from extended UD corpus (42,196 sentences)
- used the same test data (1,000) for both dependency parsing models
- compare the accuracies between two models

# 6. Experimental Setup (Cont.)

## 6.3. Evaluation

- accuracies are measured by central dependency parser performance metrics, namely, UAS and LAS

- the performance metrics are computed by:

$$UAS = \frac{number\ of\ correct\ arcs}{number\ of\ arcs}$$

$$LAS = \frac{number\ of\ correctly\ labeled\ arcs}{number\ of\ arcs}$$

# 7. Results

- higher accuracies are highlighted as bold numbers
- the models trained from the extended UD corpus are higher than the baseline models applied from the original UD corpus
- +1.09 (UAS) and +1.39 (LAS) for *jPTDP*, and +0.6 (UAS) and + 0.64 (LAS) for *UniParse*

**Table 2.** Comparison of Accuracies between Dependency Parsing Models

| Models | Original myPOS | | Extended myPOS version 3.0 | |
|---|---|---|---|---|
| | UAS | LAS | UAS | LAS |
| jPTDP | 85.07% | 81.38% | **86.16%** | **82.77%** |
| UniParse | 85.67% | 82.72% | **86.27%** | **83.36%** |

# 7. Results (Cont.)

- the extended UD corpus improves the dependency parsing accuracy
- baseline model uses a minimal amount of training data
- the extended parsing model uses approximately triple the training data applied in the baseline model
- training data of extended parsing model are taken from various domain areas
- the difference is the extended dependency parsing model can parse the raw data more accurately

# 7. Results (Cont.)

- *UniParse* model performs slightly better than the *jPTDP* model
- it is unable to parse the raw text data
- *jPTDP* model can parse the raw text data
- thus, *jPTDP* model is more useful and effective than *UniParse* model in raw text data parsing

# 8. Conclusion

- manually extended myPOS version 1.0 UD corpus to myPOS version 3.0 UD corpus
- built graph-based dependency parsers for Myanmar language
- compared the dependency parsing models built on the original and extended UD corpora
- the larger the training data, the better the model's accuracy
- the extended UD corpus is more beneficial than original UD corpus for improving the dependency parser

# 8. Conclusion (Cont.)

- We will publish the extended UD corpus and our built dependency parser as publicly available resources at https://github.com/ye-kyaw-thu/myUDTree

- In addition, we will apply the dependency information derived from our built dependency parser to factored neural machine translation

# References


[1] H. T. Z. Aye, W. P. Pa and Y. K. Thu, "Unsupervised Dependency Corpus Annotation for Myanmar Language," 2018 Oriental COCOSDA - International Conference on Speech Database and Assessments, 2018, pp. 78–83, doi: 10.1109/ICSDA.2018.8693009.

[2] D. Q. Nguyen and K. Verspoor, "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, Brussels, Belgium, pp. 81–91, 2018.

[3] D. Varab and N. Schluter, "UniParse: A universal graph-based parsing toolkit," In Proceedings of the 22nd Nordic Conference on Computa- tional Linguistics, Turku, Finland, pp. 1532–1543, 2014.

[4] E. Kiperwasser and Y. Schluter, "Simple and Accurate Dependency Pars- ing Using Bidirectional LSTM Feature Representations," Transactions of the Association for Computational Linguistics, MA, Cambridge, pp. 313–327, 2016.

[5] M. Straka and J. Straková, "Tokenizing, POS Tagging, Lemmatizing and Parsing UD 2.0 with UDPipe," In Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies, Vancouver, Canada, pp. 88–99, 2017.

[6] T. Ji, Y. Wu and M. Lan, "Graph-based Dependency Parsing with Graph Neural Networks," In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, pp 2475– 2485, 2019.

[7] X. Wang and K. Tu, "Second-Order Neural Dependency Parsing with Message Passing and End-to-End Training," In Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing, Suzhou, China, pp. 93–99, 2020.

[8] Z. Z. Hlaing, Y. K. Thu, M. M. N. Wai, T. Supnithi and P. Netisopakul, "Myanmar POS Resource Extension Effects on Automatic Tagging Methods," 2020 15th International Joint Symposium on Artificial Intelligence and Natural Language Processing (iSAI-NLP), pp. 1–6, 2020.

[9] Z. Z. Hlaing, Y. K. Thu, T. Supnithi and P. Netisopakul, "Improving Neural Machine Translation with POS-tag features for low-resource language pairs," Heliyon, vol. 8, August 2022. https://doi.org/10.1016/j.heliyon.2022.e10375

[10] B. Prachya and S. Thepchai, Technical Report for The Network-based ASEAN Language Translation Public Service Project, "Online Materials of Network-based ASEAN Languages Translation Public Service for Members," 2013.

# THANK YOU