UNSUPERVISED DEPENDENCY PARSING FOR MYANMAR LANGUAGE

HNIN THU ZAR AYE

UNIVERSITY OF COMPUTER STUDIES, YANGON

DECEMBER, 2020

Unsupervised Dependency Parsing for Myanmar Language

Hnin Thu Zar Aye

University of Computer Studies, Yangon

A thesis submitted to the University of Computer Studies, Yangon in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

December, 2020

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research and has not been submitted for a higher degree to any other University or Institution.

…..…………………………… …………........…………………………

Date Hnin Thu Zar Aye

ACKNOWLEDGEMENTS

First and foremost, I would like to thank Ministry of Education for giving me the opportunity to study PhD Courses by providing support that allowed me to perform the research in University of Computer Studies, Yangon, Myanmar.

Secondly, I would like to very special thanks to Dr. Mie Mie Thet Thwin, the Rector, the University of Computer Studies, Yangon, for allowing me to develop this thesis and giving me general support during period of my study.

I would also like to special thanks to Dr. Tin Htar Nwe, Professor, Faculty of Computer Science, University of Information Technology, for her insightful comments, suggestions, and challenging questions which inspired me to widen my research from various perspectives.

It is a pleasure to thank my supervisor, Professor Win Pa Pa, Natural Language Processing Research and Laboratory, Faculty of Computer Science, University of Computer Studies Yangon, for her patience, motivation, constant encouragement, helpful guidance and suggestions throughout my PhD research.

I would like to thank my co-supervisor, and advisor, Dr. Ye Kyaw Thu, National Electronic & Computer Technology Center (NECTEC), Thailand, for his guidance, suggestions, knowledge sharing for my research.

I would like to express my gratitude and special thanks to Professor Khine Moe Nwe, Course-coordinator of PhD 9th Batch, for her support and encouragement throughout all stages of the research.

I further would like to thank Professor Khin Mar Soe, Head of Natural Language Processing Research and Laboratory, Faculty of Computer Science, University of Computer Studies Yangon, for her support and encouragement throughout my research.

I would also like to thank Professor Mu Mu Myint, Head of Department of Information Technology Supporting and Maintenance, and Professor May Aye Khine, Head of Faculty of Computing, University of Computer Studies Yangon, for their comments, questions and suggestions in my seminar presentations.

Moreover, I would like to express my special thanks to Daw Aye Aye Khine, Associate Professor, Head of English Department for her patience and valuable suggestions which help to improve English language applied skill and the thesis writing.

I would like to thank all of my PhD 9th Batch classmates and colleagues from the NLP Lab, and my beloved friends for their motivations, knowledge sharing, constant caring, and unforgettable memories we have made in the last five years.

Lastly a very special thanks and gratitude goes to my parents and my sisters and brothers who gave me their love, kindness, patience, physically and mentally supports, and constant encouragement along the way of my life.

ABSTRACT

Parsing natural language is an important intermediate step for natural language processing field of any language and any natural language applications such as machine translation, information extraction, text analytics, and speech recognition systems. Parsing is examining the structure of sentence in terms of relationships between phrases or words of sentence and can be carried out by syntax, or constituency, or phrasal parsing and dependency parsing. Dependency structure is simpler and better than syntax or phrase structure to represent language semantic and syntactic information.

Dependency parsing provides directed links of the connection of linguistic unit (words) in sentence. Dependency structures and parsing have been more applied in natural language applications such as machine translation, and provide better performance results. Motivated research areas of unsupervised dependency parsing from raw sentence without requiring any annotated resources have achieved a big improvement in fifteen years ago and some resources and annotated treebanks of some languages have been shared to improve multilingual parsing purposes. As a result, unsupervised dependency parsing becomes a probable way to obtain dependency information of low or under resource languages and more applied.

Myanmar language has free word order nature, many styles for sentence writing, and no resource for dependency information. Therefore, it is still cost- and time-consuming, and difficult to add manually dependency structures of Myanmar words. According to these issues, this dissertation is the first proposed work for dependency parsing based on transition-based dependency parsing method that uses transition predictions of neural network classifier for Myanmar language. An adaptable Myanmar POS tag scheme which is related to Universal part-of-speech (U-POS) tags and dependencies has been also firstly defined and proposed to apply unsupervised dependency parsing. Myanmar dependency treebank has been annotated to build Myanmar parsing model to parse Myanmar sentences. Evaluation experiments of the new Myanmar parsing model have been executed. The proposed dependency parsing method has parsed well new Myanmar test sentences. Accuracies scores of experiments and evaluations of parsing performance are measured by undirected attachment score (UAS) and label attachment score (LAS). Most UAS and LAS result

scores of parsing experiments and evaluations are over 89% and 84% in general. Accuracies scores and result trees are acceptable.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

Natural language parsing is an analysis of sentence structure and a critical role of Statistical Machine Translation (SMT) and is still challenging with ambiguity and inefficiency knowledge of words. It is also an important intermediate stage for semantic analysis in natural language processing (NLP) applications such as information extraction (IE) and question answering (QA) and machine translation (MT). Parsing can be executed by two parsing techniques, by constituents (phrases) or drawing links of individual words (dependency grammar).

Dependency parsing is to find the sentence structure by directed dependency links between words in sentence. It is also suitable to deal with languages having word orders being relatively free. Some NLP applications such as machine translation, textual entailment recognition and relation extraction and question answering have well applied dependency formalisms in recent years. Dependency parsing can be performed by rule-based approach by grammar driven rules such as content free grammar rules or data driven approach by machine learning based approaches which might be supervised or unsupervised learning method.

Supervised Parsing needs a treebank, a large annotated corpus to train a statistical model. Building treebank or corpus annotation costs expensive and very time-consuming task and needs linguistic human annotators.

1.1 Unsupervised Dependency Parsing

Unsupervised parsing is independent of language theories and universal across languages. The main advantage of unsupervised parsing is that any annotated treebank is not required in unsupervised parsing. The main disadvantage is producing quite poor quality so far. However, to improve better quality, unsupervised dependency parsing approaches allow using some types of data and amount of knowledge about them by different ways according to the degree of (un) supervision.

Two Computational Natural Language (CoNLL) shared tasks have been released for dependency parsing to work from raw text over many different languages by learning syntactic dependency parsers and Universal Dependencies (UD) with cross-linguistically consistent grammar annotation. It can provide an inventory of dependency relations being motivated, computationally useful, and cross-linguistically applicable in linguistic research areas.

Treebanks also called corpora are collections of texts, where sentences are annotated with a specific syntactic configuration by syntax or constituent or dependency structures.

1.2 Problem Statements

Being free word order nature of Myanmar language, formal Myanmar sentences and colloquial sentences have been written with many different writing styles.

For one main significant case in writing Myanmar sentences, most Myanmar nouns have been usually suffixed with post positional markers (PPM)s in formal sentences while most nouns have been written without suffixing post positional markers in most colloquial sentences. Moreover, having agglutinative language nature, verbs or adjectives are usually suffixed particles and PPMs.

Moreover, some words might be used as main adjectives or verbs or as suffixed particles. Therefore, same words might be the different part-of-speech (POS) according to the content of sentences.

As an example case can be seen in the sample verb phrases. For example, in verb phrase, "**ထား သည်**" , means "**ထား**" means put and POS is verb. In a verb phrase, "**လေ့လာ ထား ပါ**" , "**ထား**" shows the state of making order or requesting politely to do the main action verb, "**လေ့လာ**" , means study and it's POS is particle. As an one more example, in sample verb phrase ,"**လုပ် ထား သည်**" , "**ထား**" shows the state of already finishing the action verb, "**လုပ်**" , means do.

Therefore, it is difficult to identify dependencies between words in sentences as the possibility of different part-of-speech (POS) tags for the same word. POS information is important to identify word dependencies.

1.3 Motivation of the Research

Myanmar language is free word order nature and still low resource language. Developing dependency treebank needs annotators to develop and check dependency structures and is hard and time- and cost-consuming task but syntactic dependency information resource is a critical resource and important for natural language processing in languages. Moreover, dependency parsing is more suitable and useful to represent syntactic information for languages having free word order nature than widely used syntax parsing in natural language parsing. Therefore, the first and main motivation of this research is to build dependency syntactic information resource, dependency treebank, for Myanmar language.

Unsupervised dependency parsing can provide better qualities of induced structures than before while supervised parsing can provide high accuracy and is widely used. There is a great improvement in language research areas of dependency parsing by unsupervised methodologies ten years ago. Even though the result qualities of pure unsupervised parsing approaches are poor, current unsupervised approaches let to use specific types of data or knowledge about them depending on the different level of supervised or unsupervised state to get better results. However, unsupervised dependency parsing can support to obtain syntactic structure for low resource languages without using any dependency annotated resource and any knowledge.

Being the recent progresses of unsupervised dependency parsing, the second motivation of this research is to apply unsupervised dependency parsing in building dependency corpus, treebank, to provide fast and easily linguistic semantic information to be able to reach statistical approaches and deep learning by bootstrapping way for Myanmar language.

Currently, being lack of the parsing model resource for Myanmar sentence parsing, third motivation is to build a dependency parsing model to be able to parse directly Myanmar sentences by Myanmar model because dependency information of natural language sentences is useful for natural language processing applications like statistical machine translation.

1.4 Objectives of the Research

The main purpose of this research is to implement unsupervised dependency parsing architecture for Myanmar language parsing. And the other related objectives of this research are as follows:

1. To understand syntax structures of Myanmar sentences
2. To implement parsing model in order to be used as a basic parsing model for Myanmar Language
3. To build Myanmar dependency treebank
4. To apply unsupervised transition-based dependency parsing based on transition prediction by Neural Networks classifier model for parsing Myanmar sentences

1.5 Contributions of the Research

The procedures of the first proposed unsupervised dependency parsing for Myanmar language and contributions of this research are as follow.

There are two main parts: training and testing in order to build unsupervised dependency parsing for Myanmar language.

In training part, data preprocessing was carried out as a main process before training parsing model. One important part for data preprocessing is that it is needed to use unique schemes in segmentation and part-of-speech tagging on sentences of different corpora which are input and used in Myanmar dependency parsing

To be able to use unique schemes in segmentation and part-of-speech tagging, a general POS tag scheme related to Universal part-of-speech (U-POS) tags is contributed to tag with same POS tags on sentences of different Myanmar corpora used in dependency parsing.

Then a mapping scheme between Myanmar POS and Universal POS in sentences is contributed to easy adding Universal POS tags in sentences.

In data preprocessing, firstly Universal part-of-speech tags of input sentence is added. Then, unsupervised annotating was applied by Japanese shared model provided by UD project by using UDPipe pipeline process tool to add raw dependency structures since most Japanese grammar structures are similar to Myanmar grammar structures. Manual annotation for all dependency structures of words in Myanmar sentences takes a long time and needs annotators. Unsupervised annotating raw dependency structures via the shared model of other language by using UDPipe tool of UD project is contributed to fast annotating dependency heads and relations structures for Myanmar language being low resource language without using many human annotators.

As one main contribution of this research, validation of unsupervised raw annotation results by the referenced dependency structures was carried out in manual post processing by one annotator after unsupervised annotation to be reliable dependency information for Myanmar sentences.

After post processing, main contribution and the last step of training part of this research, Myanmar dependency parsing model was built to use as a basic model for parsing Myanmar sentences written by various styles from domain areas.

When training part has finished, a parsing application has been built to test and evaluate the training model, for unsupervised dependency parsing based on transition predictions of Neural Networks classifier model to parse Myanmar sentences. This parsing application based on transition predictions of Neural Networks classifier model is the first parsing application for Myanmar language and one contribution of this research as well.

1.6 Organization of the Research

This dissertation is organized with seven chapters. This chapter describes the brief introduction of the research with motivations, objectives and contributions of the research.

Chapter 2 presents the summary of previous works related to parsing and dependency information of Myanmar language, related works of unsupervised dependency parsing and building universal dependencies treebanks.

Chapter 3 provides the issues of Myanmar sentence parsing and the background theories and algorithms of dependency parsing, types and approaches of dependency parsing, and universal part-of-speech tags and universal dependencies.

Chapter 4 presents the procedures of building dependency treebank for Myanmar language.

Chapter 5 demonstrates how to parse sentences by transition-based dependency parsing based on transition predictions of neural network classifier to get dependency parse trees for Myanmar sentences.

Chapter 6 explains the results of experiments studied in this research and discussion of the analysis and evaluations on those results.

Chapter 7 describes conclusion and the future works of this research.

CHAPTER 2

LITERATURE REVIEW

This chapter describes the summarized review on related works of dependency information structures of Myanmar sentences and related works of parsing Myanmar sentences for Myanmar language. First related works of parsing Myanmar sentences were based on function tagging and grammar rules by top down parsing method. The next related works based on the dependency structures of Myanmar sentences used annotated resources.

In addition, CoNLL shared tasks will be briefly described. And Universal Dependencies (UD) project and development tools for UD for multilingual parsing project for across languages, and building treebanks with UD standards of some languages will also be described.

2.1 Related Works of Myanmar Sentence Parsing

Since Myanmar sentences have free-phrase-order and complex morphological system, some early works for Myanmar language processing have been proposed by rule-based approaches or supervised methods based on machine learning approach.

A parsing approach was proposed by using context free grammar (CFG) and assigning function tags for Myanmar sentences [68]. The proposed method consists of two main parts: function tagging and parsing by context free grammar (CFG) [68].

2.2 Parsing Myanmar Sentences with Function Tagging

Function tagging was used as a pre-processing step for parsing. Function tags are useful to extract and learn more about the behavior of words in sentences. Function tagging is a kind of process to assign syntactic roles of each word in sentence like subject, object, time, location, etc., to provide useful semantic information of sentences.

Function tagging process of the previous work used theory of Naive Bayes and the functionally annotated tagged corpus [68]. In this proposed method, total 39 function tags in which one tag for verb phrase and 38 tags for other phrases were defined according to Myanmar grammar which consists of 17 kinds of post positional marker (PPM). Most function tags are identified by combination of word and suffixes, post positional marker (PPM) type, based on grammar structure types of post positional marker (PPM).

Naive Bayesian based classifier was proposed for the model of function tagging due to a Naive Bayes classifier can be developed for learning how much each function tag should be trusted for its decision makings [37].

The proposed system works at word-level and input sentences are pre-segmented, part-of-speech-tagged and chunked. The proposed function tags will be recognized as a class types and task of words of input sentence. Sample input sentence and output of the function tagging process will be illustrated in Figure 2.1 for an example sentence. A sequence of word-tags of the input sentence was represented as "noun conjunction ppm noun ppm noun ppm verb". Then, it was also necessary to add a sequence of chunk as "NC CC NC PPC NC PPC NC PPC VC SFC". Sentence-(a) of Figure 2.1 is input segmented, POS tagged sentence with related chunks. Words of input sentence were added related POS tags to obtain more accurate lexical information in order to be formed as the features of words. Sentence-(b) is output with function tags.

Sample Input Sentence: "သူတို့သည် မောင်ဘကို ခေါင်းဆောင်အဖြစ် ရွေးချယ်ခဲ့သည် ။"
In English : "They chose Mg Ba as a leader."

(a) NC[သူတို့/pron.possesive]#PPC[သည်/ppm.subj]#NC[မောင်ဘ/n.person]#PPC[ကို/ppm.obj]#NC[ခေါင်းဆောင်/n.person]#PPC[အဖြစ်/part.eg]#VC[ရွေးချယ်/v.common,ခဲ့/part.support]#SFC[သည်/sf]။
(b) Psubj[သူတို့]#SubjP[သည်]#Pobj[မောင်ဘ]#ObjP[ကို]#PPcmplO[ခေါင်းဆောင်]#PcomplOP[အဖြစ်]#Active[ရွေးချယ်ခဲ့သည်]။

Figure 2.1 Sentence structures defined in previous work

Parsing is analyzing a sequence of words, tokens, of a text or sentence to recognize the grammatical structure of it according to a given grammatical rules. Parsing of the proposed method is to generate parse trees of function tagged Myanmar sentences by predefined context free grammar (CFG) rules. The positions of phrases in sentences are not fixed since Myanmar language has free phrase order nature.

In Myanmar grammar, there are two sentence types called as simple and complex. A simple sentence contains one verb or adjective. A complex sentence contains two or more simple sentences joined by postpositions, or particles, or conjunctions.

Therefore, it is impossible to define all types of structures of Myanmar sentences and only some simple types of sentence structures had been made grammar rules in that work. Example sentences types defined in simple grammar rules are shown in Figure 2.2.

သူ-သည်-ကျောင်း-သို့-သွား-သည်။	(Subj-Pla-Verb)
သူ-သည်-ကျောင်းသားတစ်ယောက်-ဖြစ်-သည်။	(Subj-PcomplS-Verb)
ကောင်စီဝင်-အဖြစ်-သူ့-ကို-လူထု-က-ရွေး-သည်။	(PcomplO-Obj-Subj-Verb)
မောင်လှ-သည်-ခွေး-ကို-တုတ်-ဖြင့်-ရိုက်-သည်။	(Subj-Obj-Use-Verb)
သူ-သည်-ဆရာ့-ကို-စာအုပ်-ပေး-သည်။	(Subj-Obj-Iobj-Verb)
သူမ-သည်-လူနာများ-ကို-ဆွေမျိုးများ-ကဲ့သို့-ပြုစု-သည်။	(Subj-Obj-Sim-Verb)
ကလေးများ-သည်-အဖော်-ကြောင့်-ပျက်စီး-သည်။	(Subj-Cau-Verb)
သစ်ရွက်တို့-သည်-တပေါင်းလ-၌-ကြွေ-သည်။	(Subj-Tim-Verb)
တရားသူကြီး-သည်-ခိုးမှု-ကို-တရားရုံး-၌-နံနက်-က-စစ်ဆေး-သည်။	(Subj-Obj-Pla-Tim-Verb)
အမေသည်-သူ့သားအတွက်-မုန့်ကို-**ဈေး**မှ-မနက်က-ဝယ်ခဲ့သည်။	(Subj-Alm-Obj-Pla-Tim-Verb)

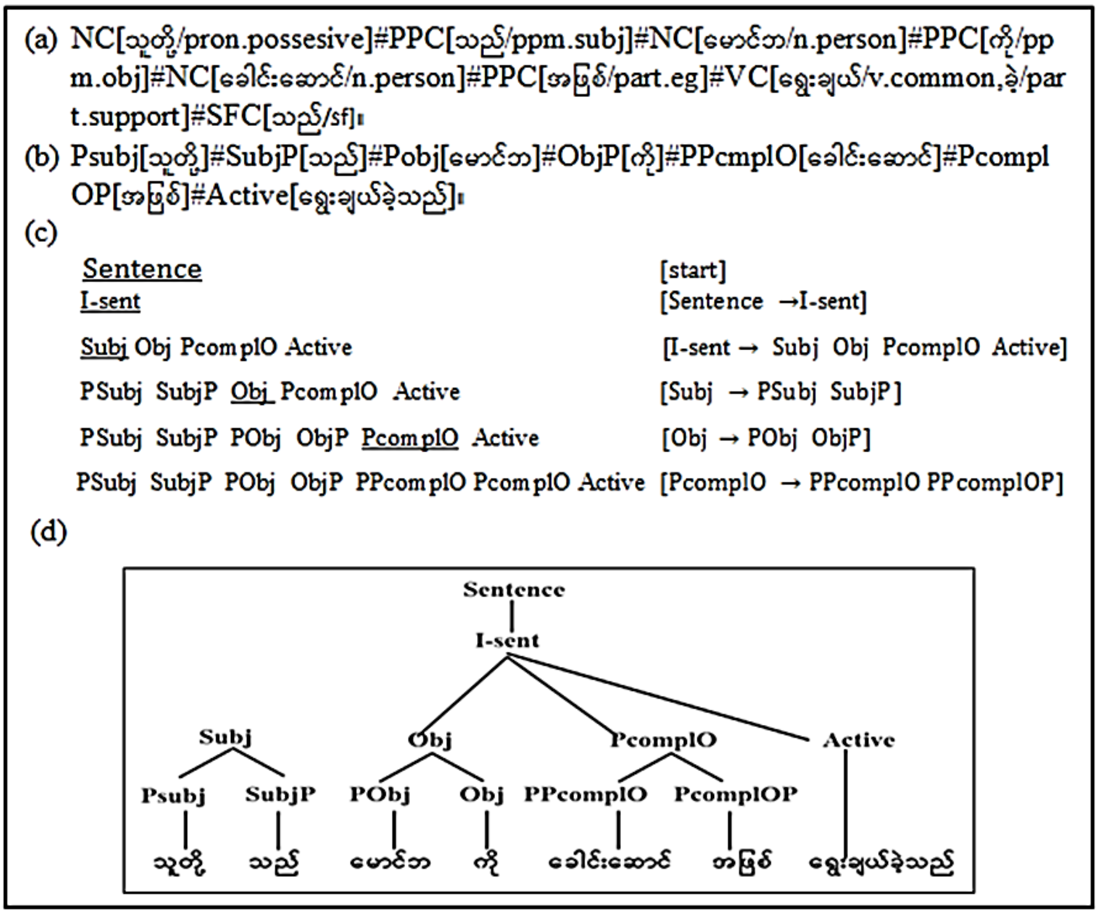
Figure 2.2 Example sentence types defined in previous work

Therefore, the grammar rules had not been considered for all sentences types in the previous proposed work and could not parse for all Myanmar sentences types [2-1]. The example proposed grammar rules defined to parse simple sentence structures are shown in Figure 2.3.

```
Sentence   →  I-sent|I-sent CC I-sent | CCM I-sent | Obj-sent I-sent | Subj-sent I-sent
I-sent     → Subj Obj Pla Active | Subj Active | Com Pla Active | Subj PcomplS Active
CC         → CCS | CCP
Subj-sent  →  I-sent CCA Subj
Obj-sent   →  I-sent CCA Obj
Subj       →  PSubj  SubjP
Subj       →  Subj
Obj        →  PObj ObjP
Obj        →  Obj
Pla        →  PPla PlaP
PcomplO    →  PPcomplO PcomplOP
Use        →  PUse UseP
Sim        →  PSim SimP
```

Figure 2.3 Content free grammar of the previous work

CFG rules were proposed for grammatical relations of function tags in the previous work. The proposed CFG grammar could generate simple types and simple forms of three complex types of Myanmar grammar sentence structures according to predefined related CFG grammar rules for types of related sentence structures. The example flow of work and output format for parsing input simple sentence are shown as Figure 2.4.



(a) NC[သူတို့/pron.possesive]#PPC[သည်/ppm.subj]#NC[မောင်ဘ/n.person]#PPC[ကို/ppm.obj]#NC[ခေါင်းဆောင်/n.person]#PPC[အဖြစ်/part.eg]#VC[ရွေးချယ်/v.common,ခဲ့/part.support]#SFC[သည်/sf]။

(b) Psubj[သူတို့]#SubjP[သည်]#Pobj[မောင်ဘ]#ObjP[ကို]#PPcmplO[ခေါင်းဆောင်]#PcomplOP[အဖြစ်]#Active[ရွေးချယ်ခဲ့သည်]။

(c)

Sentence	[start]
I-sent	[Sentence →I-sent]
Subj Obj PcomplO Active	[I-sent → Subj Obj PcomplO Active]
PSubj SubjP Obj PcomplO Active	[Subj → PSubj SubjP]
PSubj SubjP PObj ObjP PcomplO Active	[Obj → PObj ObjP]
PSubj SubjP PObj ObjP PPcomplO PcomplO Active	[PcomplO → PPcomplO PPcomplOP]

(d)

Figure 2.4 (a) POS tagged chunks (b) Function tagging (c) Derivations (d) Parsed tree examples of previous work

The training corpus used in that work was built by total 3900 sentences in which 1,600 are simple sentences having no more than 15 words and 2300 are complex sentences having more than 15 words. Sentences collected from the middle school Myanmar language textbooks and the historical books written in Myanmar had been tested for evaluation. That test data set contained about 2,200 sentences.

The proposed system could produce 96.68%, 93.05%, and 94.83% for precision, recall, and f-measure scores respectively for the 670 correct sentences of 693 recognized sentences from 720 test sentences for simple sentence type. For testing complex sentences, the proposed system could produce:

- 93.81 % of precision, 88.54% of recall, and 91.09% of f-measure for the 394 correct sentences of 420 recognized sentences from 455 test complex sentences joined with postpositions,
- 90.88 % of precision, 86.22% of recall, and 88.48% of f-measure for the 319 correct sentences of 351 recognized sentences from 370 test complex sentences joined with particles,
- 92.66 % of precision, 89.17% of recall, and 90.88% of f-measure for the 593 correct sentences of 540 recognized sentences from 665 test complex sentences joined with conjunctions [68].

The previous proposed method had been considered for limited sentence types of Myanmar to construct the CFG grammar rules. Therefore, it is needed to consider for other sentence types for more function tags and the grammar rules to parse since the phrase orders of Myanmar sentence is relatively free. Moreover, that system could not control for some particular sentence types since unlimited word positions might be occurred in other sentences.

However, they could describe a corpus expandable method by the outputs of the function tagging model of Naïve Bayesian classifier to build larger functional annotated corpus for Myanmar to English translation system in order to reduce time consuming in corpus creation based on experimental results scores. In addition, output function tagged sentences could parse successfully by defined production rules of content free grammar for related grammatical rules for Myanmar phrases.

The second related work is a context free grammar (CFG) based top-down parsing approach to parse function tagged Myanmar sentences [67]. A context-free grammar can define a language by a set of derivable strings derived from a starting symbol called as sentence symbol. A CFG has four main tuples <N, T, P, S>. N represents a set of non-terminals and T represents a set of terminals. P is a set of production rules and S is a sentence or start symbol, non-terminal symbol. Top down parsing is goal one strategy that build parse from the start Symbol (S). Top Down parsing is goal oriented parse a sentence from starting symbol, S, according to production rules of the grammar and repeats derivation steps until the parse tree matches with the input terminal string.

Writing a CFG grammar for all sentence types of any natural language is so hard and also for Myanmar language. Therefore, they designed a grammar to generate simple Myanmar sentences by the production rules of CFG for only simple grammars in order to write simply production rules of CFG. That CFG was built by function tags and consists of 38 rules for function tags that were combined to get the phrases and 183 rules for grammatical relations of the phrases. Training corpus was built with about 3,000 sentences for three simple types and three complex types of sentences. Trained corpus was evaluated by 530 test sentences for those six types of sentence structures for the performance of the defined CFG rules for Myanmar sentences. Average accuracy score had been reported with 90.6% for six types of sentences [67].

2.3 Related works on Myanmar Sentence Dependency Structures

Based on the dependency natures of Myanmar language, a dependency -based head finalization approach had been proposed for statistical Machine Translation (SMT) experiment and a parsing approach for Myanmar sentences by using Japanese similar syntactic structures to Myanmar as a pivot.

2.3.1 Dependency-based Head Finalization

There are similar syntactic structures especially in sentence order, Subject-Object-Verb (SOV), and suffixing post markers between Myanmar, Japanese, Chinese and Korean.

Based on these dependency syntactic structures, a simple dependency-based head finalization scheme for the purpose of statistical machine translation (SMT) was proposed for Myanmar language. In that proposed approach, head finalization with a head–driven phrase structure grammar (HPSG) for English-to-Japanese translation were combined with dependency-based pre-ordering scheme originally designed for English-to-Korean translation to perform the phrase-based (PB) SMT system in Moses [4].

The corpus called Basic Travel Expression Corpus (BTEC) [19] was used in their experiments. Chinese (zh), English (en), French (fr) were source languages and Myanmar is used as a target language Myanmar (my). In the experiments 155,121 sentences, 5,000 sentences, and 2,000 sentences had been used for training, development, and test data respectively for each source and target languages.

In experiments with Moses [49], the phrase-based (PB) SMT system was used as a baseline system. GIZA++ [17] was used to align word and symmetrized alignment was used by grow-diag-final-and heuristics [48]. The lexicalized reordering model was trained with the msd-bidirectional-fe option [8]. The maximum phrase length was 7. For Myanmar training data, SRILM [1] was used to training 5-gram language model with interpolated modified Kneser-Ney discounting [58].

The default settings of the Moses decoder were adopted except the distortion-limit (DL) in the process of decoding. That is, table-limit was 20 and stack is 200. As DL value setting, 0, 6, and 12, and ∞ were used in experiments. The parameter weights on the development sets were tuned by MERT [18] and the translation of test data sets were evaluated by two automatic measures: BLEU [34] and RIBES [21].

Experimental results had shown that the proposed head finalization approach is able to attain higher performance than unsupervised baseline SMT result without parallel training data.

2.3.2 Parsing Approach by Statistical Machine Translation

A parsing approach was proposed based on mapping shared similar syntactic structures between Japanese and Myanmar chunks to parse Myanmar sentences [5]. Parsing by chunks and dependency relations mapping on intermediate Myanmar-to-Japanese translated chunks sequences resulting from statistical machine translation (SMT) of Myanmar sentence could be carried out as the following steps:

- first, translation input Myanmar sentence into Japanese by SMT
- parsing translated Japanese sentence into chunks
- mapping dependency relations between intermediate similar Japanese chunks and Myanmar to parse Myanmar sentence.

The basic travel expressing corpus (BTEC) [19] was used to investigate the performance of their proposed approach. From BTEC corpus, 457,249 sentences, 5,000 sentences and 3,000 sentences were used for training, development, and test data set respectively.

For the PB SMT, MOSES [49] was used with default settings in training and decoding. The language model used in SMT was an interpolated modified Kneser-Ney discounting 5-gram model, and trained on the Japanese part of the training set by

SRILM [670]. The Myanmar sentences were segmented into words by an in-house CRF-based segmenter tool for the SMT system. MeCab with IPA dictionary for segmenting and CaboCha were used for chunking and parsing [65] for the Japanese output sentences from the SMT system.

Experimental results for chunk and dependency accuracy had received over 95%. The chunk and dependency structures of a Myanmar sentence could be mapped from its Japanese translation with the help of a PB SMT system and a Japanese parser. Experimental results on BTEC corpus were examined manually and illustrated the proposed approach could perform satisfactory. The high numerical results could demonstrate that the proposed approach actually generated reasonable paring results on Myanmar sentences. The unknown Myanmar words did not affect the approach much on the translated Japanese sentences, because the Japanese parsing method more relies on those functional morphemes, which are usually translated well[5].

2.4 CoNLL Shared Tasks

A shared task is featured in the conference on computational natural language learning (CoNLL) each year. A shared task on multilingual dependency parsing had been emerged in the tenth CoNLL (CoNLL-X). How to convert treebanks of 13 languages into the same dependency format and how to measure parsing performance as the shared task were described in 2006 [57]. During ten years before 2006, there were much research done on parsing by data-driven machine learning approach and performance results have increased steadily.

Treebanks, syntactically annotated corpora, consisting thousands to tens of thousands sentences have been necessary to train those parsers. For different treebanks data, various annotation schemes and logical data formats have been used and had been tedious to apply a parser to many treebanks. The shared task introducing a uniform approach to dependency parsing had been expected to improve that situation. 19 participant groups had participated in that shared task. It had been taken two to three months to implement a parsing system that could be trained for all these languages and four days to parse unseen test data for each.

The training data from the original treebanks given to the shared task format was a simple column-based format that is an extension of MALT-TAB format of Joakim Nivres. All sentences will be in one text file and they are separated by a blank line after a sentence. A sentence contains one or more tokens. Each token will be represented as one line, consisting of 10 fields. Each field will be separated by a TAB. The 10 column fields are:

(1) ID for token counter, starting at 1 for each sentence.

(2) FORM for word or punctuation symbol.

(3) LEMMA for lemma or stem (depending on the type of treebank) of word, or an underscore if not available.

(4) CPOSTAG for coarse-grained part-of-speech tag, where the tagset depends on the treebank.

(5) POSTAG for fine-grained part-of-speech tag, where the tagset depends on the treebank. It will be same as the CPOSTAG value if no POSTAG is available from the original treebank.

(6) FEATS for unordered set for syntactic and/or morphological features (depending on the particular treebank), or an underscore if not available. Set members are separated by a vertical bar (|).

(7) HEAD for head of the current token, which is either a value of ID, or zero ('0') if the token links to the virtual root node of the sentence. Note that depending on the original treebank annotation, there may be multiple tokens with a HEAD value of zero.

(8) DEPREL for dependency relation to the HEAD. The set of dependency relations depends on the particular treebank. The dependency relation of a token with HEAD=0 may be meaningful or simply "ROOT' (also depending on the treebank).

(9) PHEAD for projective head of current token, which is either a value of ID or zero ('0'), or an underscore if not available. The dependency structure resulting from the PHEAD column is guaranteed to be projective (but is not available for all data sets), whereas the structure resulting from the HEAD column will be non-projective for some sentences of some languages (but is always available).

(10) PDEPREL for dependency relation to the PHEAD, or an underscore if not available.

A wide variety of parsing approaches were used. In addition, systems were scored by computing the labeled attachment score (LAS), i.e. the percentage of scoring tokens for each system's prediction the correct head and dependency label. Although results across languages and systems had varied widely from 37.8% (worst score on

Turkish) to 91.7% (best score on Japanese), variations are consistent enough to draw some general conclusions [57].

In addition, in 2007 as in 2006, the Computational Natural Language Learning Conference featured a shared task in which participants had trained and tested their learning systems on the same data sets. In 2007, the shared task had been devoted to dependency parsing, with both a multilingual track and a domain adaptation track. Moreover, how to define tasks of the different tracks and how to create data sets from existing treebanks for ten languages were also described. Then the test results were also reported and an analysis on those results was also provided. In 2007 shared task, the multilingual track with an annotated training and testing data from a wide range of languages were processed with one and the same parsing system to be able to learn from training data, to generate to unseen test data, and to handle different languages, possibly by adjusting hyper-parameters [28].

In data-driven parsing systems , needing the adaption parsers from plentiful resources domains to little resource domains is important and is referred to as domain adaptation for the purpose which is able to adapt annotated resources from a source domain to a target domain . The setup scheme of 2007 shared-task for domain adaption case is to use no annotated resources in the target domain since the research work by McClosky et al. [14] and Blitzer et al. [22] had revealed that the existence of a large unlabeled corpus in the new domain can be leveraged in adaptation. It was also tested in English by providing large annotated corpus from Wall Street Journal news as the source domain and data from biomedical abstracts assumed as development data, chemical abstracts assumed as test data set 1, and parent-child dialogues assumed as test data set 2 by assuming as three different target domains.

English, Czech, Italian, Hungarian, Greek, Chinese, Catalan, Basque, Turkish, and Arabic treebanks were used in 2007 shared task for multilingual and domain adaption tracks and sizes of train, development, and test data set for each language were also reported. And the evaluation scores for the multilingual track had been measured and reported by label attachment and unlabeled attachment scores by languages of that treebankes in the shared task of CoNLL 2007. In addition, the evaluation scores of eleven teams for the experiments of domain adaption track in English had been measured and reported by label attachment and unlabeled attachment scores.

The two main paradigms of system architectures of each participant, transition-based parsers and graph-based parsers had been analyzed in 2007 CoNLL shared task. Moreover, building parsing model, inference technique and learning for inference in two paradigms had been described.

In 2006 and 2007 CoNLL shared task, dependency parsing could convey a huge enhancement to the growth of multi-language dependency parsers and also to some level for multi-domains.

Increasing the multiple connections between annotation scheme, language structure, and learning methods, and parsing become the most important direction to consider for future research in multi-language dependency parsing area. However, the outputs of two CoNLL shared tasks could constitute a potential milestone for comparative error analysis in across languages and systems. And they are also freely accessible for research.

2.5 Universal Dependencies Project

In Universal Dependencies (UD) is an international cooperative project that is creating cross-linguistically consistent treebank annotation for many languages, aiming to facilitate multilingual parser development, and learning and parsing across research from perspective types of a particular language.

The annotation scheme is based on an advancement of Stanford dependencies [41] [40] [43], universal part-of-speech tags of Google [62], and the tagset conversion method that is reusable and, to a reasonable extent, universal to be able to interpret part-of-speech and morphological tags in a uniform way [15].

The general idea of UD is to provide sets and rules of a universal inventory to facilitate consistent annotation for similar constructions across many languages.

2.5.1 History of Universal Dependencies

The Stanford dependencies were developed in 2005 to help in Recognizing Textual Entailment systems as a backend to the Stanford parser. Then they were ultimately emerged as the de facto standard for dependency syntactic analysis of English, and have since been adapted to some different languages [50] [2] [32] [38] [56] [53].

The universal tag set of Google was raised out of the error analyzing linguistically of different languages based on the CoNLL-X shared task data [54]. It has been initially used for unsupervised part-of-speech tagging [11], and has since been accepted as a usual standard for mapping various tag sets to a common standard.

The first combining Stanford dependencies and Google universal tags into a universal annotation scheme became the Universal Dependency Treebank (UDT) project [42], which could release treebanks for 6 languages in 2013 and 11 languages in 2014, and the first proposal for incorporating morphology was made [56] .

The second version of HamleDT [55] provided Stanford/Google annotation for 30 languages in 2014. This was followed by the development of universal Stanford dependencies (USD) [44].

Merging all those initiatives into a single comprehensible framework based on Stanford universal dependencies, an extended version of Google universal tagset, a revised subset of the interset feature inventory, and a revised version of the CoNLL-X format known as CoNLL-U format, a new dependencies framework known as Universal Dependencies has emerged .

2.6 Building Universal Dependencies Treebanks

Universal dependencies have been proposed for many languages in UD cooperative project and some have been shared. The Universal Dependencies (UD) project emerged to develop multilingual parser, cross-lingual learning, and parsing research from a language typology perspective. The UD project has been developing cross-linguistically consistent treebank annotation for many different languages for multilingual parsing.

A new method collecting treebanks with the homogeneous syntactic dependency annotation scheme had been presented based on six languages; English, French, German, Swedish, Spanish, and Korean in order to facilitate multilingual syntactic analysis research by showing usefulness of that resource in case studying the cross-lingual transfer parsing with reliable evaluation. It is freely available. And it has been extended in order to take account of more data and languages.

Moreover, building dependency treebanks with Universal dependencies for many languages have been described. Dependency treebanks are increasing within a few years. There are over 100 treebanks of more than 70 languages available in the

UD inventory up to February 2019. (At the present time (February 2019). Some of them will be described briefly in this section.

A Japanese word dependency corpus annotation with word dependency annotation standard guideline based on manual word segmentation has been presented. It was annotated nearly 30 thousand sentences collected from six different domain sources such as questions and answers from web site, sentences on blog site, books, magazines, newspaper articles, dictionary sample sentences, and patent disclosure of machine translation task. It is also compatible with the high quality and widely used Balanced Corpus of Contemporary Written Japanese (BCCWJ) for various NLP tasks and enough to train statistical parsers for general domain. The preliminary parsing experiments had been done by MST-based parser on annotated corpus to evaluate the usage of it in building parser and have been reported [61].

Universal Dependencies syntactic annotation scheme for Japanese had been presented to build Japanese UD corpus with mapping schemes to Universal part-of-speech and syntactic annotation in UD [66].

In 2018, building Universal Dependencies (UD) 2.0 version resources for Japanese has been presented. The UD Japanese resources are built by automatic conversion from several treebanks. Several Japanese treebanks with different word delimitation, POS, and syntactic relations could be ported for the UD annotation scheme. Moreover, issues in word delimitation, POS and syntactic relations in Japanese case markers, clauses, and coordinate structures have been described [66].

In order to affiliate Romanian Treebank to Universal Dependencies, building dependency treebank with manual and automatic manually checked annotation had been presented. The syntactic relationships of that treebank were **meticulously** defined. An annotation interface could be built form that treebank creation and a dependent parser for Romanian language could also be built and it can work with statistical methods [7].

Building gold-standards for the dependency grammar for Norwegian had been described with the way of choosing data format to annotate, and morphological and syntactic annotations [51].

The annotation and parsing by ensemble support vector machine (SVM) method had been introduced for Indonesian dependency treebank [46].

For ancient Greek and Latin language, dependency treebanks have been developed from the large scale collections of classical texts including words with explicit syntactic, morphological and lexical information. Ancient Greek dependency treebank was composed of different sentence types and contained total 21,170 sentences with 309,096 words. Latin dependency treebank was composed of different sentence types and consisted of total 3,473 sentences with 53,143 words. Moreover, the influence of the usage of that treebanks in a cultural heritage digital library was presented and could take advantages in reading environment with canonical standards for the presentation of text and a large body of digitized resources including XML source texts, morphological analyzers, machine-readable dictionaries, and an online user interface [9].

A Turku dependency treebank for Finnish has been presented as a publicly available. The treebank contains total 15,126 sentences with 204,399 tokens manually annotated in a Finnish specific version of the Stanford Dependency scheme from 10 different text sources. And the morphological analyzing has been assigned by a novel machine learning method to disambiguate readings given by an existing tool. And the first open source Finnish dependency parser could be presented as an open source and trained on the new annotated treebank. The parser could generate achieve the labeled attachment score of 81 %. The treebank data are also available under an open license at http://bionlp.utu.fi/ [33].

Deriving three dependency Korean treebanks from existing corpora and pseudo-annotated by the latest UD version 2 guidelines have been presented to convert phrase structure trees across different treebanks into dependency trees with consistent relations, providing a large corpus of compatible dependency trees. Three Korean corpora could be converted together into dependency trees following the latest UD guidelines by automatic conversion by using head-finding rules and linguistic heuristics and contains total of 38K+ dependency trees [23].

2.7 Chapter Summary

In this chapter, the previous research works relating to Myanmar sentences parsing have been briefly described. And also the proposed works based on the dependency structures of Myanmar sentences have been presented. In addition, the related works of unsupervised dependency parsing on across multi-languages have been presented. Moreover, universal dependencies and universal dependency project doing researches for the purpose of multi-lingual parsing process on across languages have been explained briefly. Building the treebanks with universal dependencies in some language has been described.

CHAPTER 3

DEPENDENCY PARSING

This chapter presents tree parsing methods, dependency parsing techniques and algorithms. Moreover, universal part-of-speech tags, universal dependencies, dependency treebank are also presented. Building dependency treebanks with universal dependencies are briefly described. The issues of Myanmar sentences parsing are also presented.

3.1 Issues of Myanmar Sentences

Myanmar language is agglutinative and morphological rich language. Basic Myanmar sentence structure is Subject-Object-Verb (SOV). The grammatical hierarchy is useful for successively included levels of grammatical construction operating within and between grammatical levels of analysis for Myanmar sentences. Figure 3.1 presents the grammatical hierarchy of Myanmar sentences.

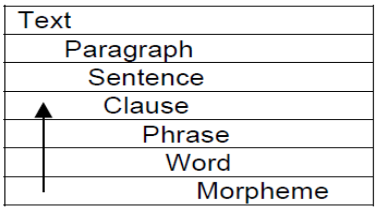
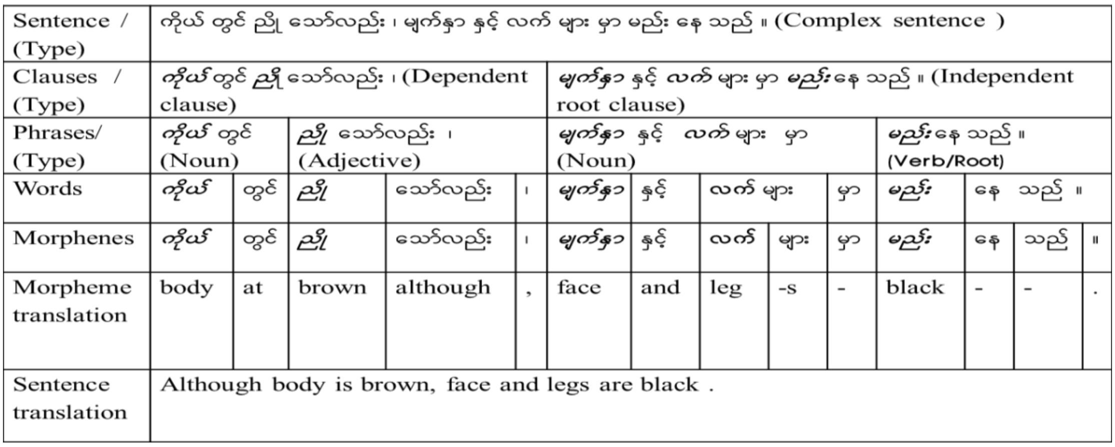


Figure 3.1 Grammatical hierarchy structure of Myanmar sentence

Myanmar sentences writing style can be formal or colloquial. Generally, there are two main sentence types in construction of Myanmar sentences as simple and complex type. Simple sentences can be composed of a noun phrase and a verb phrase. Complex sentence has two or more simple sentences or clauses which are connected by a conjunction, or post positional marker, or particle to provide modified or extra meaning for the followed part which might be clause or phrase of main combined sentence.

In Myanmar language, noun phrases are written with different types of postpositional markers (PPMs), nominal markers in sentences as formal styles to describe their roles such as object, subject but noun phrases in colloquial style sentences are not written with these markers. Myanmar sentences might be composed of one or more clauses or phrases. To provide more detail meaning for modified clauses or phrases, some subordinate clauses are commonly used and they are also placed before modified ones. Nested cases of this condition become time-consuming task in defining dependency relations between main and sub parts of Myanmar sentences. Example sentence structure can be seen in Figure 3.2.



Sentence / (Type)	ကိုယ် တွင် ညို သော်လည်း ၊ မျက်နှာ နှင့် လက် များ မှာ မည်း နေ သည် ။ (Complex sentence)													
Clauses / (Type)	***ကိုယ်*** တွင် ***ညို*** သော်လည်း ၊ (Dependent clause)					***မျက်နှာ*** နှင့် ***လက်*** များ မှာ ***မည်း*** နေ သည် ။ (Independent root clause)								
Phrases/ (Type)	***ကိုယ်*** တွင် (Noun)		***ညို*** သော်လည်း ၊ (Adjective)			***မျက်နှာ*** နှင့် ***လက်*** များ မှာ (Noun)					***မည်း*** နေ သည် ။ (Verb/Root)			
Words	***ကိုယ်***	တွင်	***ညို***	သော်လည်း	၊	***မျက်နှာ***	နှင့်	လက် များ		မှာ	***မည်း***	နေ သည် ။		
Morphenes	***ကိုယ်***	တွင်	***ညို***	သော်လည်း	၊	***မျက်နှာ***	နှင့်	လက်	များ	မှာ	***မည်း***	နေ	သည်	။
Morpheme translation	body	at	brown	although	,	face	and	leg	-s	-	black	-	-	.
Sentence translation	Although body is brown, face and legs are black .													

Figure 3.2 Grammatical hierarchy structure in sample sentence

There are four phrases in Figure 3.2. If adjective is attached with post positional marker suffix, it becomes verb as root verb phrase of the above example sentence in Myanmar language. The morphemes in Figure 3.3 are nominal markers in example sentence.

All Nominal markers:	"တွင်"	, "မှာ"	, "များ"
In English	: at/in,	at/in	, -s (plural marker)
Tag Types	: post positional markers		, particle

Figure 3.3 Nominal markers of example sentence in Figure 3.2

The morphemes in Figure 3.4 are suffixes of verbs. There are two types of verb suffixes: particle which is usually used to give complete meaning of action or tense of action and post positional marker (PPM) which is usually used to make verb form to followed words.

Verb Suffixes:	"နေ"	, "သည်"
In English	: -	, -
Tag Types	:particle	, post positional marker (verb marker)

Figure 3.4 Verb suffixes of sample sentence in Figure 3.2

In example sentence of Figure 3.2, main words of phrases are italic and bold. Main root phrases of a clause or sentence is final verb phrase of it. The independent root sentences are typically modified by the dependent clauses or sentences. Therefore, the right most last phrase of the example sentence is the main root of it.

In addition, most Myanmar formal sentences are regularly ended with verb phrase or verb but some sentences are not ended with verbs as the special writing style of Myanmar sentences because of hiding verbs for some actions such as being or having or coming action. There are many Myanmar sentence construction formats according to free word order nature, and some special sentence construction styles of Myanmar grammar. Moreover, according to writer's idea, emphasized noun phrases can be placed at the start of the sentence.

These conditions also become time-consuming and complicated issues in defining right dependency relations for clauses, phrases, and words in sentences besides the above-mentioned nested cases. This becomes issue in corpus annotation and treebank building.

3.2 Tree Parsing

Parsing process is to determine the sentence structure by analyzing its content words based on language grammar. Parsing is the critical role in some natural language processing applications such as machine translation, question answering, text summarizing and information retrieval where a system accept input natural language sentence and provides syntactic representation and grammatical relations of the content words of input sentence as output . In those applications, finding the linguistic structures of input natural language sentence is useful.

3.2.1 Syntax Parsing

Syntax or constituency parsing deals with grammatical arrangement of words in a sentence and their relationship with each other. Syntax parsing is extracting a constituency parse tree from a sentence with syntactic structure of sentence according to a phrase structure grammar. A constituency parse tree will break a sentence into sub-phrases. In the tree, the terminals are the words in the sentence, the edges are unlabeled, and non-terminals are types of phrases. Example syntax tree of sample sentence is presented in Figure 3.5.

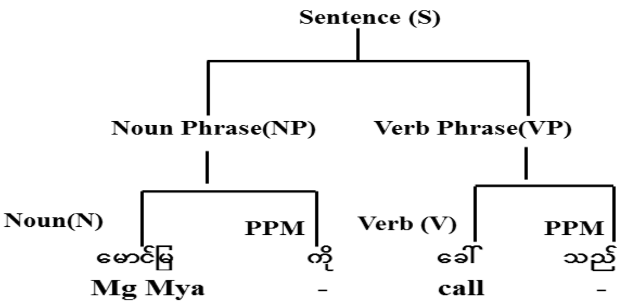


Figure 3.5 Example syntax tree

3.2.2 Dependency Parsing

Parsing natural sentences with specific structure types has been one of primary research topics in Natural Language Processing. Phrasal constituents or phrase structures rules are widely used in parsing and useful for various NLP tasks but it can be cumbersome the following three facts for free word-order languages. First, the structure is not so flexible with free word-ordering in languages such as Czech or Finnish. Second, it is somewhat language-dependent in that we need to create new sets of rules to parse sentences in different languages. Third, it is very syntax oriented in that it lacks important semantic information such as semantic roles [17].

To overcome such issues, people started focusing on a different structure called a dependency structure. There is no phrasal node in a dependency structure unlike a phrase structure. Each node in a dependency structure represents a word-token in a sentence, and except for a root node, it is dependent in exactly one other node.

Dependency parsing is the extraction a parse tree with the grammatical structure and the relationships between "head" words and words, which modify those heads of a sentence. Dependency parsing can be defined as shown in Figure 3.6.

Input :

Sentence $x = w_0, w_1, \dots, w_n$ with w_0 = ROOT

Output:

Dependency graph $G= (V, A)$ for x, where $V = \{0,1,\dots,n\}$ is the node set,

A is the arc set,i.e.,$(i,j,k) \in A$ iff $w_i \overset{k}{\rightarrow} w_j$

Figure 3.6 Definition of dependency parsing

Dependencies refer to both syntactic and semantic relations between nodes by representing as syntactic dependencies: **nmod** (noun modifier) or **nsubj** (noun subject), and semantic dependencies: **dobj** (direct object) [1]. Sample illustrations of those dependencies in example sentence are presented in Figure 3.7.

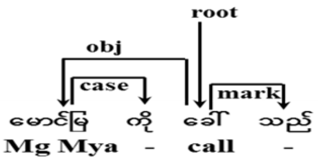


Figure 3.7 Example dependencies in Myanmar sentence

3.3 Approaches to Dependency Parsing

Dependency-based syntactic parsing systems have been performed by two main types, grammar-driven approach and the data-driven approach, although these approaches are not mutually exclusive [24].

3.3.1 Grammar-Driven Dependency Parsing

The dependency representations were intimately tied to formalizations as dependency grammar that were very close to context-free grammar. A dependency system has three rules:

1. LI: Rules of the form X $(Y_1 \dots * Y_i \dots Y_{i+1} \dots Y_n)$, where i may equal 0 and/or n, which say that the category X may occur with categories Y_1 ... Y_n as dependents, in the order given (with X in position *).

2. LII: Rules giving for every category X the list of words belonging to it (where each word may belong to more than one category).

3. LIII: A rule giving the list of all categories the occurrence of which may govern a sentence.

A sentence consisting of words $w_1 \dots w_n$ is analyzed by assigning to it a sequence of categories $X_1 \dots X_n$ and a relation of dependency d between words such that the following conditions hold (where d$*$ is the transitive closure of d):

1. For no w_i, d$*$ (w_i, w_i) .

2. For every w_i , there is at most one w_j such that d$\left(w_i, w_j\right)$.

3. If d $*$ $\left(w_i, w_j\right)$ and w_k is between w_i *and* w_j, then d $*$ $\left(w_i, w_j\right)$.

4. The whole set of word occurrences is connected by d.

5. If $w_1, \dots, w_i$ are left dependents and $w_{i+1}, \dots, w_n$ right dependents of some word, and $X_1, \dots, X_i, X_{i+1}, \dots, X_n$ are the categories of $w_1, \dots, w_i, w_{i+1}, \dots, w_n$ then X $(X_1 \dots X_i * X_{i+1} \dots X_n)$ is a rule of LI.

6. The word occurrence w_i that governs the sentence belongs to a category listed in LIII.

In the grammar-driven dependency parsing, the second main tradition is based on the eliminative parsing; where sentences are analyzed by the representation processes with successively eliminating that violate constraints until remaining with only valid representations. In that approach, parsing is viewed as a problem of constraint satisfaction, where any analysis satisfying all the constraints of the grammar is an analysis of validation. Generally, constraint satisfaction is NP complete and means that special care must be taken to ensure reasonable efficiency in practice [24].

From the view of parsing unrestricted text, parsing as constraint satisfaction can be problematic in two ways. First is that there may be no analysis satisfying all constraints, which leads to a robustness problem for a given input string. Second is that there may be more than one analysis, which leads to a problem of disambiguation.

So far, two main trends in grammar-driven dependency parsing have been distinguished. The first is closely related to context-free grammar based on a formalization of dependency grammar, and therefore usually restricted to projective dependency structures, using standard techniques from context-free parsing to obtain good efficiency in the presence of massive ambiguity, in particular dynamic programming or memorization. The second is based on a dependency grammar formalization in terms of constraints, not necessarily limited to projective structures, where parsing is viewed as a constraint satisfaction problem addressed using eliminative parsing methods, although the exact parsing problem is often intractable[24].

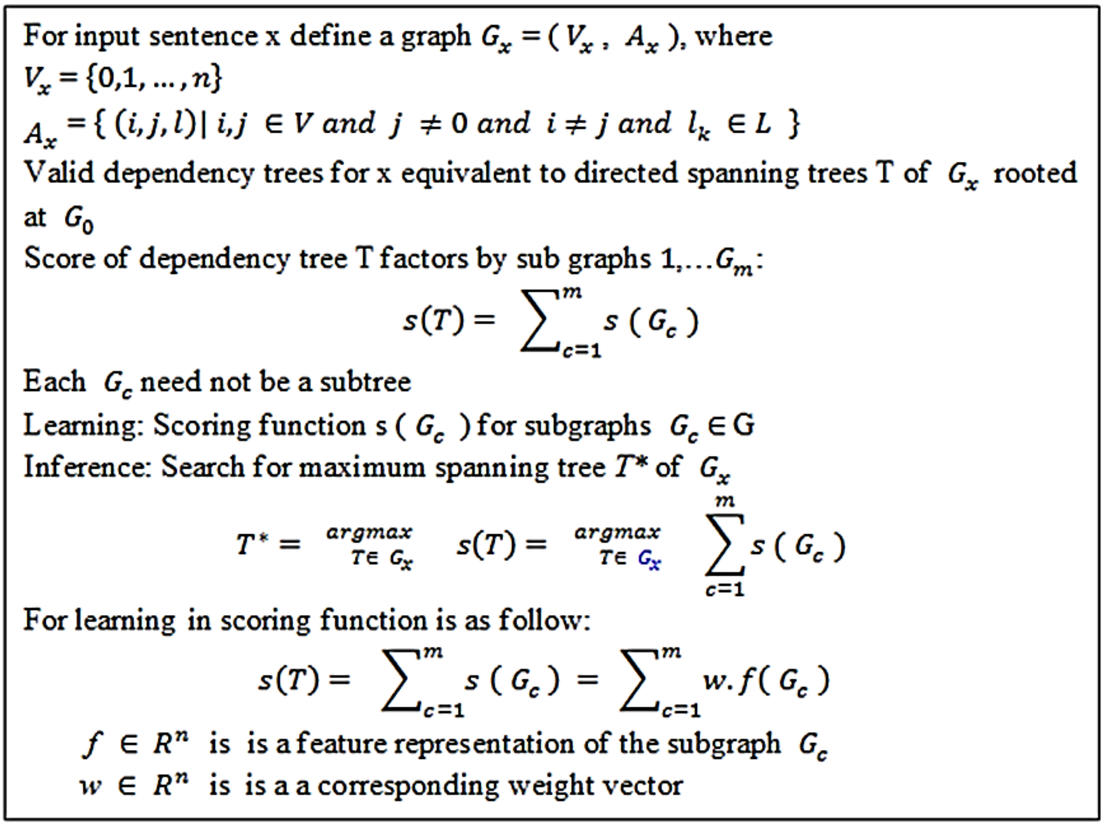
3.3.2 Data-Driven Dependency Parsing

As for natural language parsing in general, the first attempts at data-driven dependency parsing were also grammar-driven in that they relied on a formal dependency grammar and used corpus data to induce a probabilistic model for disambiguation [24].

Two main approaches of data-driven dependency parsing are graph-based and transition-based algorithms. A graph-based algorithm finds the highest scoring parse tree from all possible outputs, scoring each complete tree for a given input sentence, while a transition-based algorithm builds a parse tree based on a sequence of actions, scoring each action individually.

3.3.2.1 Graph-Based Dependency Parsing

Graph-based dependency parsers start with a completely connected graph whose edges are weighted according to a statistical model. They then try to find the spanning tree that covers all nodes in the graph (the words) and at the same time maximizes the sum of the edges belonging to the spanning tree.



For input sentence x define a graph $G_x = (V_x\ ,\ A_x)$, where

$V_x = \{0,1,\dots,n\}$

$A_x = \{ (i,j,l) |\ i,j\ \in V\ and\ j\ \neq 0\ and\ i \neq j\ and\ l_k\ \in L\ \}$

Valid dependency trees for x equivalent to directed spanning trees T of G_x rooted at G_0

Score of dependency tree T factors by sub graphs 1,...G_m:

$$s(T) = \sum_{c=1}^{m} s\ (G_c)$$

Each G_c need not be a subtree

Learning: Scoring function s (G_c) for subgraphs $G_c \in$ G

Inference: Search for maximum spanning tree T^* of G_x

$$T^* = \underset{T\in\ G_x}{argmax} \quad s(T) = \underset{T\in\ G_x}{argmax} \quad \sum_{c=1}^{m} s\ (G_c)$$

For learning in scoring function is as follow:

$$s(T) = \sum_{c=1}^{m} s\ (G_c) = \sum_{c=1}^{m} w.f(G_c)$$

$f\ \in R^n$ is is a feature representation of the subgraph G_c

$w\ \in R^n$ is is a a corresponding weight vector

Figure 3.8 Definition of graph-based dependency parsing

A dependency tree is a special case of a dependency graph that spawns from an artificial root and is acyclic. Looking at a large history of work in graph theory, the best spanning tree is to find for each dependency graph. The most common form of this type of dependency parsing is called arc-factored parsing and deals with the parameterization of the edge weights. The most common tool for doing this is MST parser [45]. Graph-Based Parsing [25] can be defined in Figure 3.8. Learning will be assumed that it is solved and linear scoring plus inference allows us to use Perceptron, MIRA, etc. to find suitable weight. Parameterizing in scoring in first-order (arc-factored) model [25] can be described in Figure 3.9.

For example arc (had, effect) in Figure 3.12, **cannot** have features over multiple arcs (siblings, grandparents), valency, etc.

Scored subgraph G_c is a single arc (i, j, k)

$$s(T) = \sum_{c=1}^{m} s(G_c) = \sum_{(i,j,k) \in T} (i, j, k)$$

Often k is dropped, since it is rarely structurally relevant

$$s(T) = \sum_{(i,j) \in T} s\ (i, j)$$

$$s\ (i, j) = max_k\ s(i, j, k)$$

Figure 3.9 Scoring function of graph-based parsing

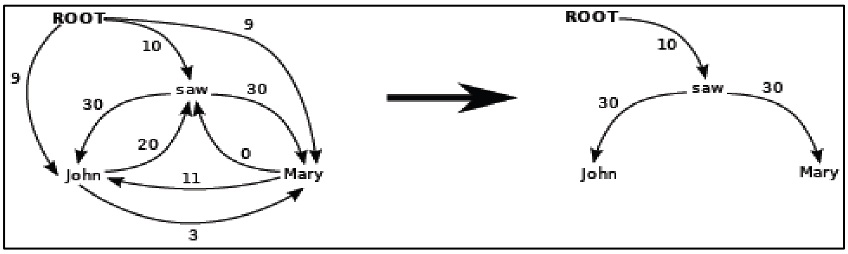


Figure 3.10 Example of sample scoring

.

f ∈ R^n is a feature representation of the subgraph G_c

For first-order models, $\boldsymbol{G_c}$ **is an arc**

i.e., $\boldsymbol{G_c = (i, j)}$ **for a head i and modifier j**

This inherently limits features to a **local scope**

Figure 3.11 Example of feature scope

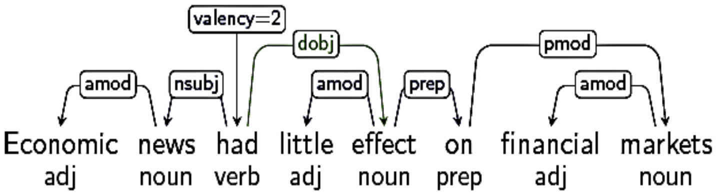


Figure 3.12 Example tree for graph-based parsing

There is trade-off in graph-based parsing for learning and inference are global especially in decoding guaranteed to find highest scoring tree and for using global structure learning in training algorithms. But that can be only possible with local feature factorizations and it is needed to limit context statistical model to look at.

In graph-based parsing, how to increase scope of features to larger sub graphs, without making inference intractable has been the major question in recent years [25].

3.3.2.2 Transition-based Dependency Parsing

Transition-based parsing creates a dependency structure that is parameterized over the transitions used to create a dependency tree. It is closely related to the shift-reduce constituency parsing algorithms. The algorithms used in these systems tend to be greedy due to the notion of picking transitions in an abstract machine. The benefit of that condition is having a linear time complexity for the algorithms. But, parses with longer arc can cause error propagation across each transition due to the greedy algorithms [60]. Malt Parser is the standard tool for transition-based algorithms [27] which in the shared tasks was often tied with the best performing systems [45].

Transition-based models for dependency parsing use a factorization defined in terms of a transition system, or abstract state machine. Different approaches are used for learning and decoding with these models: greedy classifier-based parsing and beam search, structured learning, and dynamic oracles. Most different techniques will be considered for non-projective transition-based parsing.

Transition-based Parsing have three main parts, defining a transition system for dependency parsing, learning a model for scoring possible transitions, and parsing by searching for the optimal transition sequence. **A transition system** for dependency parsing [6] can be defined as in Figure 3.13.

A transition system, S = (C, T, c_s , Ct),where:

C is a set of configurations.

T is a set of transitions, each of which is a (partial) function t: C → C.

c_s is an initialization function, mapping a sentence x to its initial configuration c_s (x).

Figure 3.13 Transition system

A configuration for a sentence can be defined as in Figure 3.14.

A configuration, c = (Σ, B, A), where:

Σ is a stack of nodes in V_x , known as the Stack.

B is a list of nodes in V_x, known as the Buffer.

A is a set of dependency arcs in $V_x \times L \times V_x$

(for some set L of dependency labels).

Figure 3. 14 Transition system

A transition sequence for a sentence x in transition systems S = (C, T, c_s , C_t) is a sequence of configurations can be described in Figure 3.15.

A transition sequence, $C_{0,m}$ = (c_0, c_1, …, c_m) of configurations:

C = c_s(x).

c_m, ∈ C_t t .

For every i (1 ≤ i ≤ m), c_i = t(c_{i-1}) for some t ∈ T .

Figure 3.15 Transition sequence

The parse assigned to x by $C_{0,m}$ is the dependency graph G_{cm} = (Vx , A_{cm}),

The dependency graph, G_{cm} = (Vx , A_{cm}), for the parse :

A_{cm} is the set of dependency arcs in cm .

More generally, the dependency graph associated with any configuration c_i for x is G_{ci} = (Vx, A_{ci}).

Figure 3.16 Dependency graph

An **oracle** is a deterministic (or may be non-deterministic) algorithm that taking a gold tree associated to a sentence, it produces the set of actions the algorithm should follow in order to reach the gold tree.

Among several transition systems, arc-standard system of Nivre will be described. Three parts in Nivre's Arc-Standard system can be presented as in Figure 3.17. It will start with a stack with the root symbol, and a buffer full of words. The terminal configuration is when buffer is empty and the stack contains only the root. The set of arcs A is not empty anymore.

Initialization:

c_s((x = x_1 , . . . , x_n) = ([0], [1, . . . , n], ∅)

Terminal Conf:

C_t = {c ∈ C|c = ([0], [ ], A) }

Transitions:

(σ, [i|β], A) ([σ|i], β, A) (**Shift**)

([σ|i|j], β, A) ([σ|j], β, A ∪ {(j, label, i)}) 1 (**Left-Arc label**)

Permitted only if i ≠ 0.

([σ|i|j], β, A) ([σ|i], β, A∪{(i, label, j)}) (**Right-Arc label**)

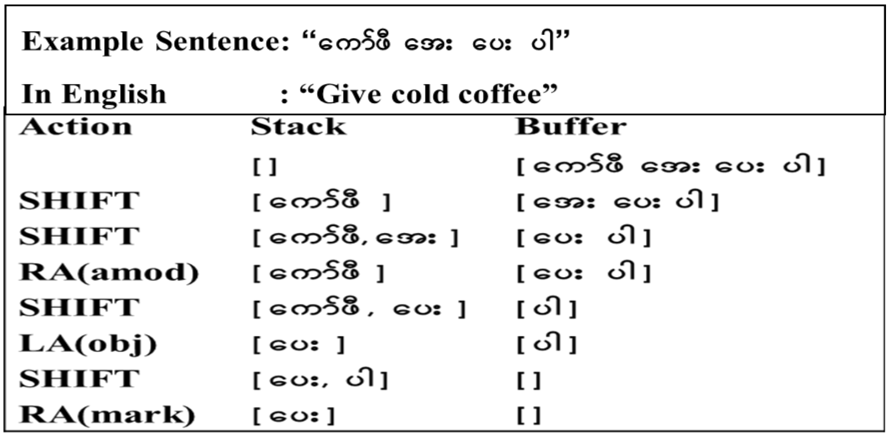
Figure 3.17 Arc standard transition system

```
$c_s$ (x = $x_1$ , . . . , $x_n$ ) = ([0], [1, . . . , n], ) ∅
while (β is not empty and !(σ has only the root))
If (σ[0] is head of σ[1] and all children of σ[1] are attached to it and σ[1] is not the root) then→Left-Arc
else if (σ[1] is head of σ[0] and all children of σ[0] are attached to it and σ[0] is not the root) then→Right-Arc
else → SHIFT
```

Figure 3.18 Configuration process of standard oracle

During the transitions, the **SHIFT** action will take the first incoming word from the buffer and push it onto the stack. The left-arc action will take the two top words at the top of the stack and creates a left-arc between them. It removes the dependent (i) from the stack. This action is permitted if the word i (the dependent) is not the root (id=0). The right-arc action will take the two top words of the stack and creates a right-arc between them. It removes the dependent (j) from the stack. In configuration process of Nivre's Standard-Oracle can be defined as in Figure 3.18.

Figure 3.19 illustrates the applied actions of the sample sentence until the last token in the stack will be attached to the root. Example dependency parsed tree is illustrated in Figure 3.20.



Example Sentence: "ကော်ဖီ အေး ပေး ပါ"

In English : "Give cold coffee"

Action	Stack	Buffer
	[ ]	[ကော်ဖီ အေး ပေး ပါ]
SHIFT	[ကော်ဖီ]	[အေး ပေး ပါ]
SHIFT	[ကော်ဖီ, အေး]	[ပေး ပါ]
RA(amod)	[ကော်ဖီ]	[ပေး ပါ]
SHIFT	[ကော်ဖီ , ပေး]	[ပါ]
LA(obj)	[ပေး]	[ပါ]
SHIFT	[ပေး, ပါ]	[ ]
RA(mark)	[ပေး]	[ ]

Figure 3.19 Applied transition actions for sample sentence

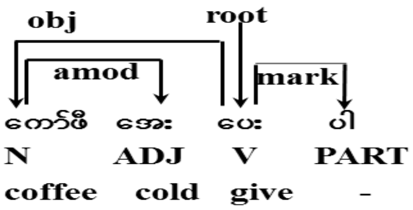


Figure 3.20 Dependency parsed tree of sample sentence

The arc-standard system considers to build a dependency tree strictly bottom-up parsing up to now:

– a dependency arc can only be added between two nodes if the dependent node has already found all its dependents.

– as a consequence, it is often necessary to postpone the attachment of right dependents.

Dependency trees have been represented by the transition sequences that derive them in a transition-based model. The transition system itself is nondeterministic (there could be several transitions that provide a solution to the same tree) and does not impose any preference among possible transition sequences for a given sentence [18].

In order to consider the optimal transition into a parser, scoring transition sequences and a method for finding the highest-scoring sequence have to be done under the model.

In the **greedy perspective**, the problem of scoring transition sequences can be reduced to the simple problem of scoring single transitions. This is also called deterministic transition-based parsing based on the local discriminative models for PCFGs as below:

$$P\big((y|x)\big) = \prod_{i=1}^{m} P\big(d_i \big| \phi(d_1, \ldots, d_{i-1}, x)\big) \tag{3. 1}$$

Parsing algorithm of greedy transition-based parsing can be defined as in Figure 3.21.

```
Parse (x = (w_0 , w_1 , . . . , w_n )
c ← c_s (x)
while c is not in Ct
t* ← argmax_t Score(c, t)
c ← t*(c)
return G_c
```

Figure 3.21 Dependency parsed tree of sample sentence

After initializing the parser to the initial configuration, the algorithm consists of a single loop that just repeatedly applies the highest-scoring transition out of the current configuration until a terminal configuration is reached [6].

The worst-case complexity of parsing is linear in the length of the sentence O(n). Given that the computation of the highest-scoring transition to the current configuration can be performed in constant time. A wide range of different models have been used to score transitions, but the most common approach is a linear model:

$$Score(c,t) = \sum_{k=1} f_k \ (c,t). w_k \tag{3. 2}$$

Greedy transition-based parsing had been first presented by [21]. Greedy transition-based dependency parsing has two problems, lack of backtracking and error propagation [18].

Beam search is a heuristic search algorithm that explores the q most promising hypotheses at each step and q is the beam size. It will be processed depending upon the value of q:

q=1 is greedy search.

- with higher q can be explored a larger part of the search space.

Beam-search requires control to a model that can score complete transition sequences:

3.4 Types of Dependency Parsing

Dependency parsing can be carried out by supervised or unsupervised methods.

3.4.1 Supervised Dependency Parsing

A parsing with annotated treebank is called as supervised parsing. Supervised parsing is based on supervised learning. Supervised parsing used supervised POS tagged and some knowledge of about the training data.

3.4.2 Unsupervised Dependency Parsing

Dependency parsing generated a parse tree with nodes and edges representing words and syntactic relations between words for an input tokenized sentence labeled by part-of-speech tags for each individual word tokens. In dependency parsing, the CoNLL shared tasks in 2006 [57] and 2007 [28] were important milestones and provided treebanks of some languages which are available in the same format. This has been used as standard to measure quality of dependency parsers up to now while there were efforts to develop a parser without needing any annotated data. A big improvement has occurred in unsupervised dependency parsing research area ten years ago.

3.5 Universal Part-of-Speech Tag Set

Part-of-Speech (POS) tagging has received a great deal of attention since it has been an important role in natural language processing systems for languages. As supervised POS tagging accuracies in English measured on the Penn Treebank have converged to around 97.3%, the attention has also shifted to unsupervised approaches. Underlying growing interest in both multi-lingual POS induction and cross-lingual POS induction via projections became the basic idea to exist a set of (coarse) syntactic POS categories in a similar form across languages.

A Universal POS tagset has been proposed to standardize best practices and facilitate future research. It consists of twelve POS categories. These categories cover the most frequent part-of-speech which also exists in most languages. Moreover, a mapping scheme from fine-grained POS tag sets of 20 different treebanks to this universal POS tag set has also developed [63]. Both the tag set and mappings are made available for down load at http://code.google.com/p/universal-pos-tags/. The universal POS categories could generalize well across language boundaries on demonstration of a grammar induction by unsupervised approach and a parser transfer task, giving competitive parsing accuracies without relying on gold POS tags [63].

3.6 Universal Dependencies

Rebuilding the standard dependency representations of Stanford, an enhanced taxonomy has been proposed to capture grammatical relations across languages, including morphologically rich ones. A proposed two-layered enhanced taxonomy is a set of broadly proved universal grammatical relations, to which language-specific relations can be added as needed. It is also important to language families or the syntax of individual languages. It has been also considered to treat grammatical relations of morphological rich languages. There are total 40 relations in the proposed taxonomy for the universal grammatical relations as listed in Table 3.1. And these relations are taken to be broadly supported across many languages in the typological linguistic literature [44].

3.7 Dependency Treebank

A treebank annotated with function dependency structures is dependency treebank. Functional annotation schemes for treebank are being interested in recent years. Dependency treebanks of many languages have been developed in recent years. In addition, many dependency-based treebanks have been constructed and annotation for grammatical functions has been also added to some constituent treebanks.

Table 3.1 Dependencies in Universal Stanford Dependencies

Core dependents of clausal predicates		
Normal dep	*Predicate dep*	
Nsubj	csubj	
Nsubjpass	csubjpass	
Dobj	ccomp	Xcomp
Iobj		
Non-core dependents of clausal predicates		
Normal dep	*Predicate dep*	*Modifier word*
Nmod	Advcl	Advmod
		Neg
Special clausal dependents		
Nominal dep	*Auxiliary*	*Other*
Vocative	aux	Mark
Discourse	auxpass	Punct
Expl	cop	
Coordination		
Conf	cc	Punct
Noun dependents		
Nominal dep	*Predicate dep*	*Modifier word*
Nummod	acl	Amod
Appos		Det
Nmod		Deg
Compounding and unanalyzed		
compound	New	goeswith
name	Foreign	
Case-marking, preposition, possessive		
case		
Loose joining relations		
list	Parataxis	Remnant
disclosed		reparandum
Other		
Sentence head	*Unspecified dependency*	
Root	dep	

The structure of dependency tree is more useful in free word order languages such as Czech, Turkish. Dependency structures are better to show relations of words in sentences while phrased-based, syntax, or constituency, or trees typically show the groups of neighboring word nodes in sentences [14].

3.8 Chapter Summary

Issues in Myanmar sentences parsing have been described with examples. And also methods in parsing natural languages have been described briefly with basic definition and algorithms. In addition, parsing techniques for dependency parsing have been presented with process procedures and sample parsing case. Universal part-of-speech tags and Universal Dependencies have been described. Treebank and dependency treebank have been described briefly.

CHAPTER 4

BUILDING MYANMAR DEPENDENCY TREEBANK

A treebank is a critical resource for natural language processing systems for languages. Dependency parsing is being more interested in natural language processing (NLP) research areas, being able to parse natural language sentences in which word order is relatively free. In many languages, dependency structures have been successfully applied in natural language application although mostly use of dependency parsing are in parsing of free word order languages. Dependency treebanks have been also constructed for many languages. Among them some have been shared for multilingual parsing project. Dependency treebank for Myanmar language has not been constructed yet. This chapter explains the procedures of building Myanmar dependency treebank for dependency parsing.

4.1 Building Myanmar Dependency Treebank

Currently Myanmar language is still low resource for syntactic information. Syntactic dependency annotation in sentences is still hard for Myanmar because of the nature of free word order. Myanmar language has generally Subject-Object-Verb (SOV) order in grammatical structure. Besides, it is also agglutinative and morphological rich. Myanmar dependency treebank has been developed by adding word dependency structures on segmented and part-of-speech (POS) tagged sentences.

4.2 Issues in Treebank Design

The treebank design is important in motivation of its intended usage for research and development of language technology. Moreover, the design of treebank is usually considered to be able to serve several purposes simultaneously since building treebank has been a very expensive and labor-intensive task based on the corpus material and the format of annotation scheme. The considerations of corpus material and annotation schemes for Myanmar dependency treebank will be presented in following sections.

4.2.1 Corpus Materials

Selecting data for a treebank is an essential consideration for corpus material. One basic fact in design choice is whether to take in written or spoken language, or both, in the treebank. The other basic consideration is whether to construct a balanced sample of different written or spoken text types or to focus on a specific text type or domain. Another issue to consider is corpus size. Treebank may be a small one with a detailed annotation or a larger one with a less detailed annotation depending on the plan of usage.

Myanmar dependency treebank contains three main corpora with three main domain sentence types: from Myanmar Wikipedia, from Web News, and from Asian Language Treebank (ALT) parallel translated news from Wiki News.

In myPOS corpus, there are 11,010 sentences which are written by colloquial and formal styles for various areas such as philosophy, history, politics, economics, and news and collected from the Myanmar Wikipedia [35].

In Web News corpus, there are 1,800 sentences which are written by most formal writing styles and collected from Myanmar news websites.

Asian Language Treebank (ALT) parallel corpus includes 20,000 sentences which are news translated from English Wiki news site [70].

4.2.2 Annotation Scheme

Most annotation schemes of treebanks are planned by layers. The lower layers are word-level annotations, such as part-of-speech, supplementing morpho-syntactic features, lemmatization or morphological analysis. Word-level annotation tends to be similar with annotation schemes of across different treebank annotation schemes.

In Myanmar treebank, word-level annotation scheme with language part-of-speech information and related universal part-of-speech information are used to be similar annotation schemes to of other treebanks of different languages.

4.3 Methodology for Creating a Treebank

A treebank can be built by automatic or manual way or both manners. Several kinds of resources such as tools or annotators are needed to build a treebank. Constructing trees manually is an error-prone process and very slow. Some treebanks have been annotated completely manually like in Norway, Finnish, and Romania languages while some have been annotated by using segmenter, tagger or parser. The most used method to develop a treebank is a semi-automatic way, combination of automatic and manual processing, but the practical implementation method varies considerably.

Building dependency treebank is a time- and cost-consuming task for any language. A big interest and progress is increasing in unsupervised dependency parsing area ten years ago. Some of the results of supervised resources have shared for other. Although result of unsupervised parsing is not good as the result of supervised method, the qualities of induced structures are better than before. However, unsupervised dependency parsing becomes possible way for low or under resourced languages to attain their syntactic structures due to multilingual unsupervised shared resources.

Being the state of the above conditions, building Myanmar dependency treebank used a combined method of manual methods and unsupervised dependency parsing as an automatic way to reduce annotated time and error-prone process.

4.3.1 Unsupervised Dependency Parsing

Unsupervised parsers infer the dependency structures based on independent properties in languages and tag set of dependency trees. Only a raw corpus should be used in an unsupervised approach. However, there are different unsupervised approaches which are having different motivations for the different degree of (un) supervision based on the fact that allows to use different kinds of data and amount of knowledge about them. In addition, to build the most probable dependency tree, current unsupervised dependency parsers were done with gold-standard supervised POS tags [13]. UDPipe project tool can be easily trained for new languages and requires neither any extra resources such as morpho-syntactic dictionaries, nor language-specific knowledge and feature engineering by the shared models of treebanks with UD of languages.

4.4 Preprocessing

Preprocessing is a prerequisite for checking and updating the word segmentation and required tagged format of each corpus before annotating treebank data with the dependency schemes to be the same format due to different corpora might be annotated with different segmentation styles and POS tag schemes in line with their original building purposes.

Preprocessing can provide an easier way to transform dependency corpus rather than carrying on different formats. The works and procedures taken as preprocessing in word segmentation and POS tagging for Myanmar treebank will be discussed in following sub sections.

4.4.1 Word Segmentation

As morphological rich and agglutinative nature of Myanmar language, most Myanmar phrases such as noun, verb, and adjectives are usually written with suffix or pre-fix words in sentences. Using word-level annotation for sentences in Myanmar treebank data, it is needed to separate main words and suffix or prefix words with the same format in sentences. An illustration of sample word segmentation in following sample sentence can be seen in Figure 4.1:

Sentence :	ကစားကွင်း၌ ကစားနေကြသည်					
Phrases :	ကစားကွင်း၌	ကစားနေကြသည်				
Words :	**ကစားကွင်း**	၌	**ကစား**	နေ	ကြ	သည်
Tag Types :	Noun	PPM	Verb	Particle used for continuous action (like – ing)	Particle used for group action	PPM (verb marker)
In English :	playground	in	play	-	-	-
Translation:	(They are) Playing in playground.					

Figure 4.1 Example of word segmentation

4.4.2 Challenges of Word Segmentation

Containing three different corpora created by each separate purpose in treebank, to be unique word segmentation format in all corpora of treebank is important. Applying unique styles in word segmentation and POS tagging can provide easy and fast dependency structure annotation.

4.4.3 Part-of-speech Tagging

For dependency information of word, POS information of each word and POS tagging scheme are important. To have a consistent POS tag scheme of Myanmar language for all Myanmar sentences, a general POS tag scheme is designed for Myanmar language specific tag scheme. In addition, using Universal-part-of-speech (U-POS) tags to apply unsupervised dependency parsing to get automatic raw annotated data, a mapping scheme between U-POS tags and Myanmar language specific tags is also needed.

4.4.4 Defining a Part-of-Speech Tag Set

Ten main types of POS tags are defined for Myanmar language in Myanmar grammar book published by Myanmar Language commission. They are noun, pronoun, adjective, adverb, post-positional marker, particles, conjunctions, interjection, and punctuation [16].

In order to be easily mapped to Universal-part-of-speech, 16 POS tags for

Table 4.1 Specific POS tags of Myanmar language

POS Tag	Description	Example (Translation)
N	Noun	သတင်း (News), အခန်း (Room), ရုံး(Office)
PRPN	Proper Noun	ဒီဇင်ဘာ (December), အာဖရိက (Africa)
NUM	Number	၅ (5), ၆ (6), ၇ (7), ၈ (8)
TNUM	Number text letter	ငါး (five), ခြောက် (six), ခုနစ် (seven), ရှစ် (eight)
FOR	Foreign word	Drone, Federal, Air-force, Byte
ABB	Abbreviation	ညွှန်/ချုပ် (Director-General)
PRON	Pronoun	ကျွန်တော်/ ကျွန်မ/ကျွန်ုပ် (I), ဤ (this), မည်သည့် (which)
ADJ	Adjective	သန့်ရှင်း (clean), နူးညံ့ (soft), ဆိုး (bad), ပေါ့ (light)
ADV	Adverb	မြန်မြန် (fast), ထပ် (again), ချက်ချင်း (immediately)
V	Verb	အိပ် (sleep), ဖတ် (read), ဖြစ် (be), ရှိ (has/have)
CONJ	Conjunction	နှင့် (and), နှင့်တစ်ပြိုင်နက် (as soon as), ပါက (if)
PART	Particle	ခန့် (about), နိုင်(can), ကြ (plural action marker)
PPM	Post positional marker	သည်/ က/ မှာ (nominal marker for subject), ၌ (at) , ၏ (of) , ဖြင့် (by)
PUNC	Punctuation	၊ , ။ , " , (or -LRB-,) or -RRB- , ", "
SB	Symbol	% , $
INT	Interjection	အိုး (Oh), အမလေး (Oh my god!)

Myanmar language specific tags were defined in for Myanmar treebank and the detailed descriptions are described in Table 4.1. Most of these tags were already defined in Asian Language Treebank project (ALT) except two tags: proper noun (PRPN) and text number (TNUM).

In most Myanmar segmented and POS tagged corpora, these tags were tagged as noun, "N". However, they have been defined as proper noun and number in Universal part-of-speech tag set. Therefore, we also defined and added these two tags in specific tag set of Myanmar language in order to be easy and fast mapping between U-POS tags and Myanmar language specific POS tags. Among 17 U-POS tags defined in UD specification, only 14 U-POS tags except not usually used three tags: X, DET, and AUX in Myanmar language were used in Myanmar treebank. A mapping scheme between U-POS tags and specific Myanmar POS tags is described in Table 4.2.

Table 4.2 Mapping Scheme between Universal POS tags and specific Myanmar POS tags

U-POS Tag	Description	Myanmar Language POS
NOUN	Noun	N
		FOR
PROPN	Proper Noun	PRPN
		ABB
NUM	Number	NUM
		TNUM
PRON	Pronoun	PRON
ADJ	Adjective	ADJ
ADV	Adverb	ADV
VERB	Verb	V
CCONJ	Coordinating Conjunction	CONJ
SCONJ	Subordinating Conjunction	
PART	Particle	PART
ADP	Adposition	PPM
PUNCT	Punctuation	PUNCT
SYM	Symbol	SB
INTJ	Interjection	INT

4.4.5 Challenges of POS Tagging

Some Myanmar words can also be different POS tag forms in sentences depending their usage purposes in sentences. Examples of this issue can be seen in Table 4.3, Table 4.4, and Table 4.5.

Table 4.3 Example sentence of POS tagging for verb

Sentence :	ပန်းကန်ထားပါ		
Phrases :	ပန်းကန်	ထားပါ	
Words :	ပန်းကန်	ထား	ပါ
Tag Types :	Noun	Verb	Particle used giving order to do action
In English :	plate	put	-
Translation:	Put the plate..		

Table 4.4 Example1 sentence of POS tagging for particle

Sentence :	အိမ်စာ လုပ်ထားပါ			
Phrases :	အိမ်စာ		လုပ်ထားပါ	
Words :	အိမ်စာ	လုပ်	ထား	ပါ
Tag Types :	Noun	Verb	Particle used **to give command or request to do action in advance**	Particle used in making polite request to do action
In English :	homework	do	-	-
Translation:	Do homework in adavance.			

Table 4.5 Example2 sentence of POS tagging for particle

Sentence :	စားပွဲ လုပ်ထား ပြီ			
Phrases :	စားပွဲ		လုပ်ထား ပြီ	
Words :	စားပွဲ	လုပ်	ထား	ပြီ
Tag Types :	Noun	Verb	Particle used **to express finishing state of action**	PPM (verb marker)
In English :	table	make	-	-
Translation:	Table has been made.			

The annotation was based on Universal Dependencies (UD) with the CoNLL-U format and Universal-part-of-speech tags by using "UDPipe". It is able to train easily the models for new language corpora without using additional resources, or specific language knowledge, or engineering features by the other languages shared models of

UD treebanks [39]. UDPipe project tool and models of UD treebanks shared by some languages are available at the project site.

Firstly, each corpus was annotated by unsupervised dependency parsing approach by shared Japanese trained model of Japanese UD treebank by UDPipe to get raw dependency syntactic information for Myanmar sentences since the general grammar structure of Japanese and Myanmar has been very similar. To annotate completely manually dependency syntactic structures for sentences of each corpus is time-consumed and error-prone process. Thus, unsupervised dependency parsing technique was used in order to reduce errors and annotation time in annotation process.

Then the reference dependency structures of words in phrases which most occurred in sentences were defined to be referred in manual checking and updating process of the post processing step.

4.5 Dependency Annotation Schemes

Most Myanmar verbs, adjectives, and nouns are usually written with suffixes such as particles or post positional markers (PPMs). Dependency relations are stated between main content words which are also attached the leaf words such as the function words in sentences in the annotation scheme of UD. Main parts of syntactic structures are head nodes and relation links called dependency relation labels between words. Dependency relation link arcs connect between heads and dependents words.

Table 4.6 Example sentences with hidden parts

Sentences	**Translation**	**Hidden Parts**
Words : ဘယ် မှာ **နေ** လဲ In English: Where - live -	Where do (you) live?	Subject
Words : ရန်ကုန် မှာ **နေ** တာ လား In English: Yangon in live - -	Do (you) live in Yangon?	Subject
Words : ရန်ကုန် မှာ **လား** In English: Yangon in -	(Do you live) In Yangon?	Subject and Verb
Words : ဒီ ကျောင်း ကို မည်သူ **ဖွင့်** ခဲ့ သလဲ In English: this school - who open - -	Who did open this school?	-
Words : ပညာရေး ဝန်ကြီး **ပါ** In English: education minister -	Minister of education	Object and Verb

In dependency trees, all clauses or phrases depend on the root of main clauses or sentences and the root of sentence are mostly **verb** or **adjective** in most regular sentence. However, the root of some sentences can be **particles** because their main verb constituents are hidden according to the short form of special exception case of Myanmar grammar rule. In addition, in Myanmar language, the parts like subjects, or objects, or main verbs of the first sentences are usually hidden in their following sentences of a specific paragraph or scenario. Therefore, it is important to choose correct word for main roots of sentences including hidden parts in annotation. Two example scenarios with short sentences sequences including hidden parts can be seen as follow. The root parts of sample sentences are bold letters. In the following example regular sentence, "**နေ**" is root verb which means "live".

Example Sentence: "ရန်ကုန် မှာ **နေ** တာ လား"
In English : "Do (you) live in Yangon?"

As an example of special form of hiding main verb constituents case, the root of the following example sentence is the interrogative particle, "**လား**" . That sentence follows the main sentence already including the asking the action verb , "**နေ**", in the paragraph or scenario. Therefore, main action verb is hidden as following case.

Example Sentence: "ရန်ကုန် မှာ **လား**",
In English : "(Do you live) In Yangon?"

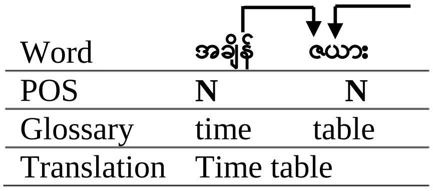
The sample main question might be as follow:

Example Sentence: "ဘယ် မှာ **နေ** လဲ"
In English : "Where do (you) live?"

A dependent word depends on head word in a phrase according to the dependency relation between words. If there are sub clauses in sentences, roots of sub clauses depend on the main root of sentences. Myanmar sentences can include zero or more clauses. Word dependency structures in phrases or clauses were defined and they are mostly occurred basic structures of noun, adjective, conjunction, adverb, and verb of Myanmar sentences. Detailed dependency head schemes of each structure will be described in the cases of compound noun, proper noun, possessive noun, numeral noun, adjective, adverb, and verb phrases, coordinate case, subordinate clauses.

Myanmar Nouns are usually written with suffixes, PPMs or particles (PART), to define their roles to the verb or specify the meaning of those nouns.

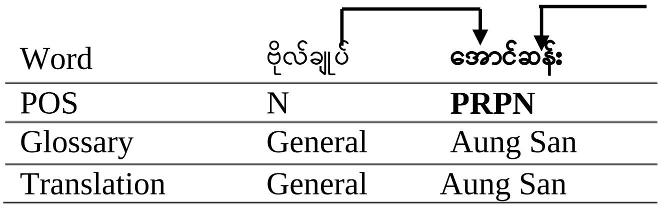
If some Myanmar words which might be nouns, or adjectives, or verb are combined, the combined phrase also becomes a new noun called, “compound noun” with another meaning to represent one syntactical role. The combined formula of example compound noun is “N N” and the dependency structure of it can be showed as Figure 4.2 in which left noun modifies right noun and the combined one is also noun.



Word	အချိန်	ဇယား
POS	**N**	**N**
Glossary	time	table
Translation	Time table	

Figure 4.2 Dependency scheme of compound noun

Proper noun is the particular unique name of common noun such as names of person, city, country, river, organization, etc. Most Myanmar proper nouns are usually used with common noun. Dependency scheme of an example proper noun phrase type is shown in Figure 4.3.



Word	ဗိုလ်ချုပ်	အောင်ဆန်း
POS	N	**PRPN**
Glossary	General	Aung San
Translation	General	Aung San

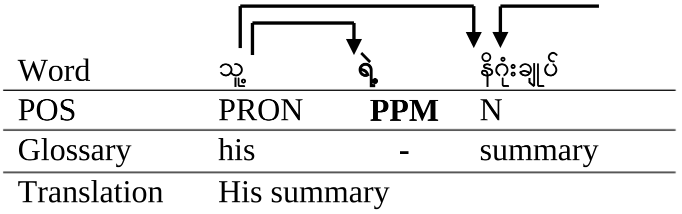
Figure 4.3 Dependency scheme of proper noun

Possessive noun phrases are written with following possessive noun suffixes by the following format as described in Table 4.7.

Table 4.7 Possessive PPMs

Possessive PPMs	**General form of possessive case**
၏ , ရဲ့ (of)	N ၏/ ရဲ့ N

. The left-most N of possessive PPM represents the owner of the right-most N which represents the substance of left. The possessive PPMs show the possessive condition of that left-most noun. Dependency scheme of an example possessive noun phrase is shown in Figure 4.4.



Word	သူ့	ရဲ့	နိဂုံးချုပ်
POS	PRON	**PPM**	N
Glossary	his	-	summary
Translation	His summary		

Figure 4.4 Dependency scheme of possessive noun

Most Myanmar **numeral nouns** are composed of main noun and number of counted amount and counting word. In Myanmar, there are two written forms to describe the number of counted amount like English as presented in Table 4.8.

Table 4.8 Myanmar numeral digits and number letters

Myanmar digits	၀,၁,၂,၃,၄,၅,၆, ၇,၈,၉
	In English: 0,1,2,3,4,5,6,7,8,9
	POS: Number (NUM)
Myanmar number letters	သုည, တစ်, နှစ်, သုံး, လေး, ငါး, ခြောက်, ခုနစ်, ရှစ်, ကိုး
	In English:zero, one, two, three, four, five, six, seven, eight, nine
	POS: Text number (TNUM)

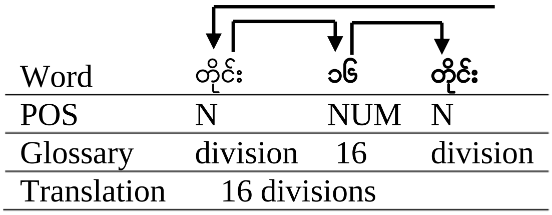
Table 4.9 Example for counting words

Counting words	**Counting things**	**Remark**
ဦး	For normal person	Numerical Particles
ပါး	For king, monk, etc.,	Numerical Particles
ခု	For unit of item like eraser, snack, etc.,	Numerical Particles
လုံး	Most rounded or circle shade items like ball, fruits, and houses and fruits	Numerical Particles
ချောင်း	for long item like ruler, pencil, stick	Numerical Particles
ကျောင်း	For school	Common noun
ကန်	For lake	Common noun
တိုင်း	For divisions or regions	Common noun

Therefore, the counted amount can be written as digits or number letters. The counting word can also be common noun of main noun or particle according to their types and shapes. The general form of numeral phrase is as follow: N NUM/TNUM

N/PART in which leftmost N represents one or more noun, NUM/TNUM is counting amount of leftmost noun which can be represented with digit number or digit letter(TNUM), and N or PART called as numerical particles. Myanmar numerical particles specify the counting amount depending on counting particles by counted noun types and shape. There are various numerical particles in Myanmar language to specify counting amount. Some example counting words are as listed in Table 4.9:

Dependency structure of an example numeral noun phrase is illustrated in Figure 4.5.



Word	တိုင်း	၁၆	တိုင်း
POS	N	NUM	N
Glossary	division	16	division
Translation	16 divisions		

Figure 4.5 Dependency scheme of numeral noun

Adjective modifies noun as in other languages. Myanmar adjectives can be before or after noun. The following particles listed in Table 4.10 are commonly suffixes to the adjectives or root verbs which specify actions or conditions of nouns and these are used to form transformed adjectives to modify nouns.

Table 4.10 Adjective suffixes

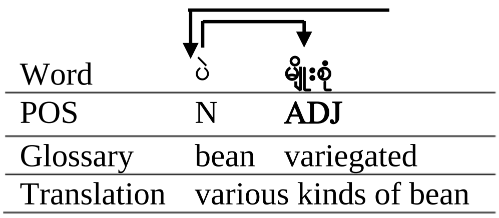
Particles	**Usage**
သော/သည့်/မည့်/တဲ့	Follow adjectives or verbs to form adjectives

Therefore Myanmar adjectives can be simple or transformed adjectives. Example of simple adjective, suffixed adjective and transformed and transformed adjective can be seen as described in Table 4.11. Both have the same meaning, “various” in English.

Table 4.11 Example of adjective types

Adjectives	**Type**
မျိုးစုံ	Simple adjective
မျိုးစုံ သော	Adjective with suffixes
အမျိုးမျိုး သော	Transformed adjective

Myanmar adjective phrases can be formed as “ADJ/V PART N” or “N ADJ”. Dependency structure of a sample adjective phrase is shown in Figure 4.6.



Word	ပဲ	မျိုးစုံ
POS	N	**ADJ**
Glossary	bean	variegated
Translation	various kinds of bean	

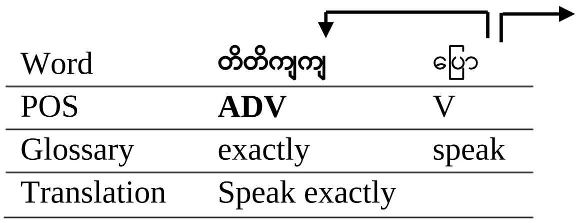
Figure 4.6 Dependency scheme of adjective

Adverb modifies the state of the verb. In an adverb phrase, there may be one or more words. If the adjective is ended with the adverb transformed particle, the combined phrase with suffix becomes transformed adverb in Myanmar language as described in the following example Table 4.12.

Table 4.12 Example of adverbs type

Adverbs	**Type**
တိတိကျကျ	Simple adverb
တိကျ **စွာ**	Transformed adverb with adverb transformed particle “**စွာ**”

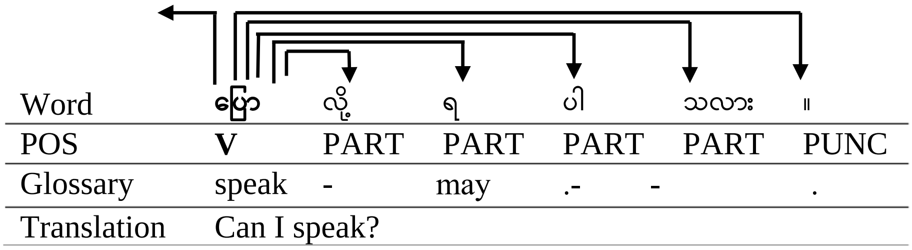
Both have the same meaning, “exactly” in English. Adverb phrases can be formed as “ADJ PART V” or “ADV V”. Figure 4.7 presents the dependency scheme of an adverb.



Word	**တိတိကျကျ**	ပြော
POS	**ADV**	V
Glossary	exactly	speak
Translation	Speak exactly	

Figure 4.7 Dependency scheme of adverb

Most **verbs** are usually ended with suffix PPMs to shape action type and tense. Moreover, zero or more particles are usually followed verbs to represent real status and tense of actions. Therefore, Myanmar verb phrases can be formed as “V PART* PPM”. Dependency structure of an example verb phrase is presented in Figure 4.8.



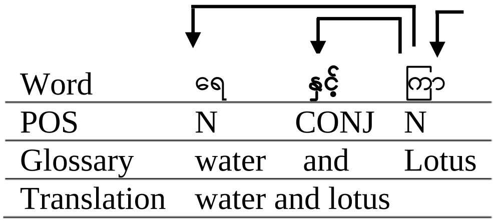
Word	ပြော	လို့	ရ	ပါ	သလား	။
POS	**V**	PART	PART	PART	PART	PUNC
Glossary	speak	-	may	.-	-	.
Translation	Can I speak?					

Figure 4.8 Dependency scheme of verb

Conjunction words are used to join phrases or sentences in order to represent coordinate meaning or join related phrases or sentences as subordinate case in Myanmar language. Example of conjunction for coordinate case and subordinate case are described as follows: Coordinating conjunction words are used to join the words or phrases or sentences. Following sample Myanmar conjunctions are used in coordinating cases by sentences or phrases or words as described in Table 4.13. Example coordinated dependency structure is presented in Figure 4.9.

Table 4.13 Example of coordinating conjuctions

Myanmar Conjunctions	**In English**
နှင့် , ရော...ပါ , ရော...ရော	and
သို့မဟုတ် . ဖြစ်စေ...ဖြစ်စေ, ဖြစ်ဖြစ်...ဖြစ်ဖြစ်	or



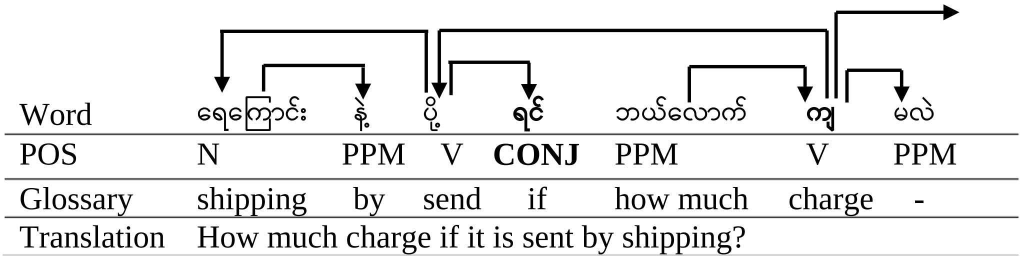
Word	ရေ	**နှင့်**	ကြာ
POS	N	CONJ	N
Glossary	water	and	Lotus
Translation	water and lotus		

Figure 4.9 Dependency scheme of coordinate case

In order to add full meaning to the main independent sentence, subordinating conjunctions are usually followed by the verbs or adjectives of dependent sentence in Myanmar language. Following sample Myanmar conjunctions are used as subordinating conjunctions as presented in Table 4.14. Example dependency structure in subordinate case is presented in Figure 4.10.

Table 4.14 Example of subordinating conjunction

Myanmar Conjunctions	In English
ရင် , လျှင်	if
နှင့်တစ်ပြိုင်နက်	as soon as
သောကြောင့် , အဘယ်ကြောင့်ဆိုသော်	because
သကဲ့သို့ , သလို	as
စေရန် , ရန် , အလို့ငှာ	in order to



Word	ရေကြောင်း	နဲ့	ပို့	**ရင်**	ဘယ်လောက်	**ကျ**	မလဲ
POS	N	PPM	V	**CONJ**	PPM	V	PPM
Glossary	shipping	by	send	if	how much	charge	-
Translation	How much charge if it is sent by shipping?						

Figure 4.10 Dependency scheme of subordinate case

4.6 Post Processing on Unsupervised Annotation

Based on modified results on unsupervised corpus annotation trees, post processing was carried out manually by referring defined dependency structures described in Chapter 4 to check trees of all sentences in the corpus aiming to build the Myanmar dependency treebank with reliable dependency trees.

Automatic unsupervised annotated results by dependency parsing of current Myanmar treebank data contain dependency head values and dependency relation labels.

As a first step of building dependency treebank, only dependency heads' values were manually post checked for correct dependency heads of each tree without updating unsupervised annotated dependency relations.

Manual post processing was done by bootstrapping way to be consistent updating and to reduce process time. To be reliable annotation and fast post processing, a new train model was built by combining updated and not updated sentences whenever we have updated selected 2,000 sentences of first corpus. Then not updated sentences were annotated again by updated trained model. Then we carried out post checking on annotated output results of these sentences and finished sentences were added to treebank.

Most main phrases occurred in sentences are proper noun, compound noun, possessive case, numeral representation, adjective, adverb phrases, verb phrases, coordinated phrases by coordinating conjunctions, and subordinating conjunction clauses. The dependencies of main sentences are also modified by subordinating conjunction clauses. Therefore, detail of word dependency structures in phrases and clauses in post processed sentences will be presented in example sentences with related explanations for their roles in sentences. Phrase dependencies are also important to analyze syntactic structure of a sentence.

The specific unique names of common nouns are called as **proper nouns** as in other languages. Myanmar proper nouns are usually found in before or after common nouns. In Figure 4.11, two proper after common nouns can be seen in example sentence. In that sentence, example **possessive case** can also be seen. Possessive condition of a noun or pronoun can be written by a post positional suffixed marker which shows that left noun of it owns right noun of it.

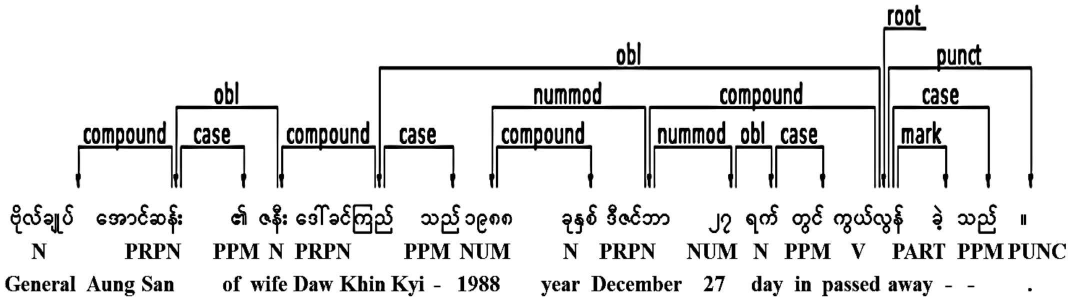

root
obl
punct
obl
nummod
compound
case
compound
case
compound
case
compound
nummod
obl
case
mark
ဗိုလ်ချုပ် အောင်ဆန်း ၏ ဇနီး ဒေါ်ခင်ကြည် သည် ၁၉၈၈ ခုနှစ် ဒီဇင်ဘာ ၂၇ ရက် တွင် ကွယ်လွန် ခဲ့ သည် ။
N PRPN PPM N PRPN PPM NUM N PRPN NUM N PPM V PART PPM PUNC
General Aung San of wife Daw Khin Kyi - 1988 year December 27 day in passed away - - .

Figure 4.11 Example proper noun and possessive case in sentence

There might be one or more words in one Myanmar noun and the POS tags of these words can be nouns, verbs, adjective or adverbs. The example sentence for the dependency structure of compound noun is shown in Figure 4.12. It contains two consecutive nouns to form a specific noun. In the example sentence, the left noun modifies the right one.

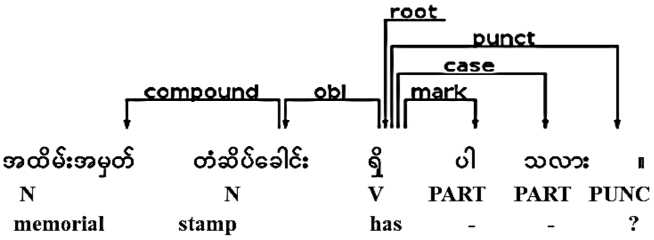


Figure 4.12 Compound noun example in sentence

Table 4.15 Example of numeral noun phrase forms

Numeral Phrase Forms	**Example**	**Meaning**	**Remarks**
N NUM/TNUM PART/N	ခဲတံ ၃/သုံး ချောင်း Pencil 3/three - (English)	3/three pencils	
N N (noun affixed particle, “အ”, to form common nouns) NUM/TNUM	ခဲတံ အချောင်း သုံးဆယ် Pencil - thirty (English)	thirty pencils	if counted amount is exact numbers of multiple of ten, hundred, thousand, etc.
N N(transformed or common noun) NUM/TNUM PART/N	ခဲတံ အချောင်း သုံးဆယ့် သုံး ချောင်း Pencil - thirty three - (English)	thirty three pencils	

Numeral noun phrases can be written mostly by the following three general formats in sentences based on the form of numeral noun forms described in above section. Examples of these can be seen in Table 4.15. The first one is used to describe the counted amount being less than ten. The counting amount can be written by text numbers or digit numbers. In Figure 4.13, sample illustration of numeral phrases in sentence is shown.

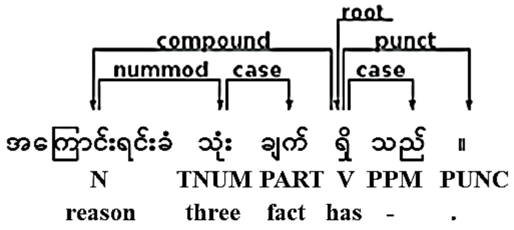


Figure 4.13 Numeral phrase in sentence

Adjectives modify nouns and they are usually placed before or after noun in Myanmar sentence. Myanmar adjectives might also be suffixed by particles. Myanmar adjective can be written by simple adjective words or simple adjective with suffix particles or transformed adjective forms. Suffixes of adjective or adjective phrases and examples of adjective forms can be seen generally as shown in Table 4.16.

Table 4.16 Adjective suffixes and example adjective forms

Particles used to suffix or transform adjective or adjective phrase		သော, သည့်, မည့်, တဲ့
Example Myanmar adjective phrases	**Translations**	**Remarks**
နီ သော ပန်း **red** - flower	red flower	Simple adjective followed suffix
ပန်း နီ flower **red**	red flower	Simple adjective
သူငယ်ချင်း ပေး သည့် ပန်း friend give - flower	flower given by friend	Transformed adjective

An example dependency structure of adjective phrase in sentence is illustrated in Figure 4.14.

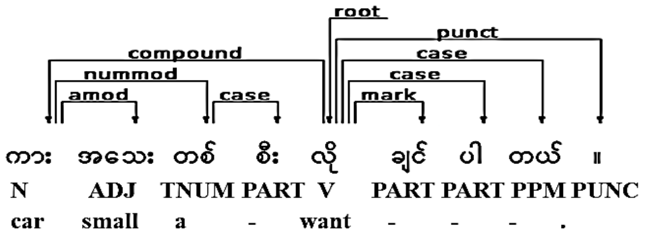


Figure 4.14 Example adjective phrase in sentence

Adverb can be written by one or more words as compound adverb or transformed adverb by suffixing particle, "စွာ" , to adverb, adjective or verb, or adverb. Example adverbs are shown with three adverb forms described in Table 4.17.

Table 4.17 Example adverb forms

Example Myanmar adverb phrases	**Meanings**	**Remarks**
နှေးကွေး (slowly)	slowly	Simple adverb
နှေးကွေး (slowly)	slowly	Simple adverb
နှေးကွေး စွာ **slow** -	slowly	Transformed adverb following suffix particle

The first two forms are usually used in both literature and colloquial style sentences. The last example is usually used in literature styles. Figure 4.15 shows an example dependency structure of adverb phrase in an example sentence.

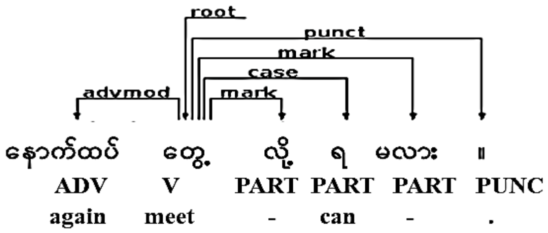


Figure 4.15 Adverb phrase example in sentence

Myanmar verb phrases are typically ended with one of the particles or the verb ended post positional markers which are shown in Table 4.18.

Table 4.18 Example of suffixes to verb

Suffixes to verb	Description of usage and example	Suffix Type
သည်, ၏, ပြီ, မည်, မယ်, တယ်	to form verb by showing tense Examples: သွား သည် (go) သွား မယ် (will go)	post positional markers
မှာ, ပါ, စမ်း	to express giving order or answering action Examples: လုပ် ပါ (giving order or requesting "Do" action) မောင်း မှာ (answering or estimating "drive" action based on the meaning of content words in sentence)	particles

. In addition, verbs can be also suffixed by zero or more particles before verb ended suffixes to describe the complete state of the action. In the example sentence of Figure 4.15 which means "Can meet again?", verb phrase sequence, "တွေ့ လို့ ရ မလား" , which means "can meet" in which "တွေ" is main verb word of that phrase and the other three words is suffixed sequence, "လို့ ရ မလား" , which means showing the polite questioning state. That main verb phrase is also modified by left adverb phrase, "နောက်ထပ်" . Therefore, the verb, "တွေ့" , is root of that sentence.

Myanmar conjunctions are used to combine not only simple sentences or clauses but also related phrases or phrases. Besides, by ending with suffixes which are particles “ဟု , သော , သည့် , မည့်, တဲ့” or post positional markers, “က, မှာ, ကို” , simple sentences or clauses can be connected to form adjective or noun clause in main sentences. If two or more simple sentences are connected by post positional markers or particles or conjunctions, combined sentence becomes complex sentence clauses represent the role such as subject or object or adjective or adverb phrases.

Most clauses connected by conjunctions are commonly adverb modifier in main complex sentence. Therefore, three clause types can be classified for the roles of dependent clauses which can be adjective clause, or adverb clause, or noun clause depending on their roles in sentences. If the connected simple sentences contain same subject, it can be omitted in dependent clause or independent clause or in both. In the same way, object can be omitted in dependent clause or independent clause if it is placed in one [16].

Myanmar conjunctions listed in Table 4.19 are used to connect sentences, phrases, or words in writing Myanmar sentences to form combined meaning [16]. Therefore, these conjunctions were tagged as U-POS tag, “CCONJ”, coordinating conjunction. Sample coordinated phrase in sentence is showed in Figure 4.16.

Table 4.19 Example of coordinating conjuction

Conjunctions	Usage
နှင့် , လည်းကောင်း…လည်းကောင်း, ရော…ပါ, ရော…ရော, သို့မဟုတ်, ဖြစ်စေ…ဖြစ်စေ, ဖြစ်ဖြစ်…ဖြစ်ဖြစ်, သော်လည်းကောင်း…သော်လည်းကောင်း, မှတစ်ပါး, ပြီး	to connect words, phrases in sentence for extra meaning
ထို့ပြင်, ထို့အပြင်, ၎င်းပြင်, သည့်ပြင်	to give connection between prior sentence and next by giving coordinated extra meaning.

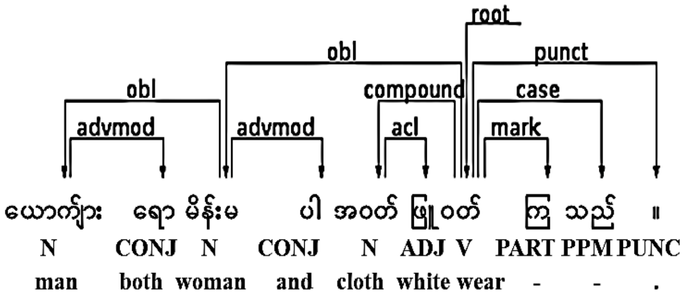

root
obl
punct
obl
compound
case
advmod
advmod
acl
mark
ယောက်ျား ရော မိန်းမ ပါ အဝတ် ဖြူ ဝတ် ကြ သည် ။
N CONJ N CONJ N ADJ V PART PPM PUNC
man both woman and cloth white wear - - .

Figure 4.16 Example coordinated phrase in sentence

Moreover, complex sentence containing two simple sentences connected by conjunction can also be seen in Figure 4.17. Left and right words of conjunction are clause or dependent sentence and independent clause respectively and subject is omitted in both.

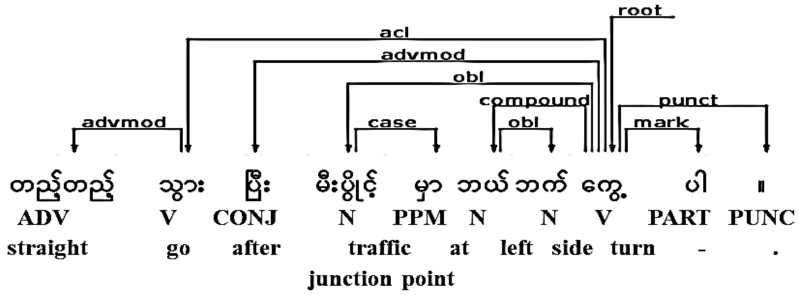

root
acl
advmod
obl
compound
punct
advmod
case
obl
mark
တည့်တည့် သွား ပြီး မီးပွိုင့် မှာ ဘယ် ဘက် ကွေ့ ပါ ။
ADV V CONJ N PPM N N V PART PUNC
straight go after traffic junction point at left side turn - .

Figure 4.17 Coordinated clause example in sentence

Myanmar conjunctions described in Table 4.20 are often used to connect clauses to provide the required meaning for main related clauses according to their meanings [16]. Therefore, they are tagged as U-POS tag, "SCONJ", subordinating conjunction.

Table 4.20 Example of subordinating conjunction

Conjunctions	**Usage**
လျှင်/ရင်, သောကြောင့်, နှင့်တစ်ပြိုင်နက်, သကဲ့သို့, စေရန်, သောအခါ, သော်လည်း,...	to connect clauses to provide the required meaning for main related clauses according to their meanings : "if, because, as soon as, as/like, in order to, when, although, etc.,"

An example complex sentence involving noun clause as an object of sentence can be presented in Figure 4.18 and it means "Lost of important documents is wanted to be reported". The PPM word "**ကို**" is used to connect the left clause to represent as object noun for the root verb, "**တိုင်ကြား**" of main sentence.

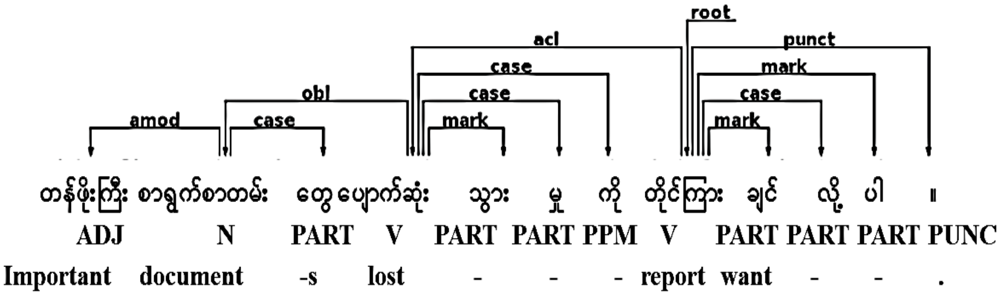

root
acl
punct
case
mark
obl
case
case
amod
case
mark
mark
တန်ဖိုးကြီး စာရွက်စာတမ်း တွေ ပျောက်ဆုံး သွား မှု ကို တိုင်ကြား ချင် လို့ ပါ ။
ADJ N PART V PART PART PPM V PART PART PART PUNC
Important document -s lost - - - report want - - .

Figure 4.18 Complex sentence including object noun clause

In addition, the next example complex sentence connecting with subordinating conjunction is illustrated in Figure 4.19 and it means “If defendant is caught, I will get in touch”. In the right main independent clause, subject is omitted.

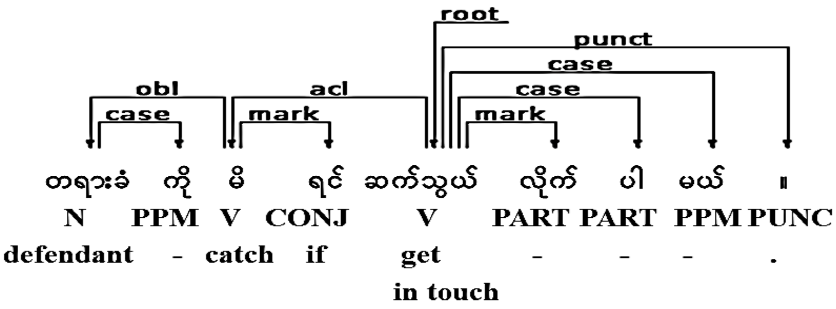

root
punct
case
case
obl
acl
case
mark
mark
တရားခံ ကို မိ ရင် ဆက်သွယ် လိုက် ပါ မယ် ။
N PPM V CONJ V PART PART PPM PUNC
defendant - catch if get in touch - - - .

Figure 4.19 Complex sentence including clause

4.6.1 Annotating Different Domain Data for Treebank

After post checking and updating 11,010 sentences of the first corpus, myPOS, finished sentences were added to treebank. After post processing and updating 11,010 sentences, the post processed data were trained again to build the updated model.

The post processed updated trained model could generate more close results to training data and post processed time of these results was faster than the former ones. As the parsed result of updated trained model were more close to referenced structures, the updated Myanmar model could automatically annotated sentences from different domain corpora, 10,000 sentences from ALT corpus and 1,800 sentences from Web News corpus, by unsupervised dependency parsing. Then 1,800 sentences of Web News among unsupervised annotated result as different domain data were post processed because post processing on Web News sentences was faster than processing on ALT sentences.

Then sentences of Web News among unsupervised annotated results of different domain data were post processed because they have more reliable sentence writing styles for the perspective of Myanmar grammar and post processing time is also faster than processing on sentences of ALT corpus in which most sentences are long sentences containing one or more sub clauses based on translated meanings from English.

4.7 Data Statistics of Updated Myanmar Treebank

The statistical information of updated Myanmar dependency treebank data [p5] after post processing is shown in Table 4.21.

Table 4.21 Statistics of Myanmar dependency treebank

Corpus Source	Domain	Sentences	Tokens	Average Length	Remark
myPOS	Wikipedia	11,010	239,534	21.75	Manually annotated
Web News	Websites' News	1,800	66,710	37.06	Manually annotated
ALT	Wiki news	10,000	373,974	37.39	Automatic annotated
	Total	22,810	680,218	29.82	

The length of sentences is also one main part of treebank data for the characteristic of involving various types of syntax and syntactic structures. For that reason, Table 4.22 presents all ranges of sentence length in treebank with corpora by classifying five levels by manual python script. The graph of sentence length range of all corpora in Myanmar dependency treebank can be seen in Figure 4.20.

Table 4.22 Classification of sentence information in treebank

Sentence Type	**Corpora**			**Total sentences of each range**	**Remark**
	myPOS	**ALT**	**Web News**		
Simple short sentence	2,253	13	74	2,340	<=10 words
Short sentence	6,463	3,858	732	11,053	>=11 and <=30 words
Normal sentence range	1,929	4,398	568	6,895	>=31 and <=50 words
Long sentence	349	1,687	401	2,437	>=51 and <=100 words
Very long sentence	16	44	25	85	>100 words
Total sentences	**11,010**	**10,000**	**1,800**	22,810	

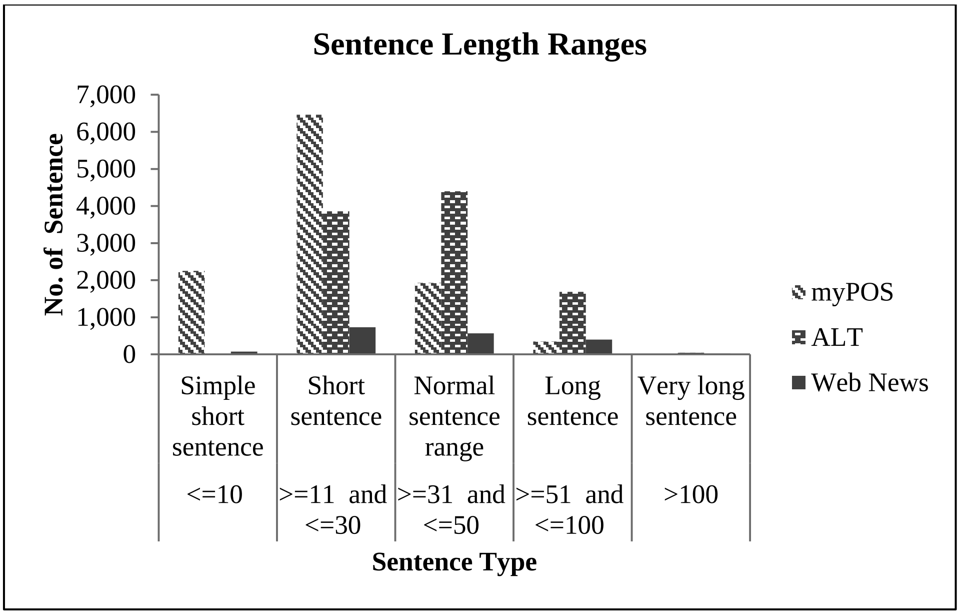


Figure 4.20 Sentence ranges in Myanmar dependency treebank

4.8 Syntax Structures of Myanmar Dependency Treebank

The detailed syntax analysis on treebank data was analyzed since syntax information has been one of the most important characteristics of treebank in order to know the syntax status of treebank. Moreover, by manual python script, syntax structures of the reference dependency types were also analyzed to count all types of dependency structures being most occurred and countable types in each sentence. To count all syntax information of natural language sentences by program script is difficult due to the issues of sentence construction types and phrases discussed in section 3.1 of chapter 3.

But, overview syntax structures of formal sentences from each corpus could be extracted by manual python script by the information of sequence order of words and their POS tag information especially for clauses and phrases written by formal literature written style.

The program counts the related phrase and clause types by checking POS tagged words order sequences in sentences having dependency structure types as described in section 4.5 in Chapter 4. But some informal phrases not ended with the formal related suffixes, PPM and particles, could not be counted by the program. Countable syntax structures [p5] are listed in Table 4.23. The state graph of phrases and clauses including in Myanmar dependency treebank is shown in Figure 4.21.

Table 4.23 Total Phrases and clauses structures in treebank

Phrase Types	myPOS	ALT	Web News
Noun	27,083	48,466	6,316
Proper Noun	10,514	17,750	3,013
Numeral Noun	5,790	10,852	2,125
Compound Noun	21,096	25,706	9,537
Adjective	9,535	11,993	1,581
Adverb	3,027	4,580	837
Verb	16,593	19,136	3,584
Phrases with Conjunctions	2,830	3,534	921
Clauses	7,027	9,837	1,950

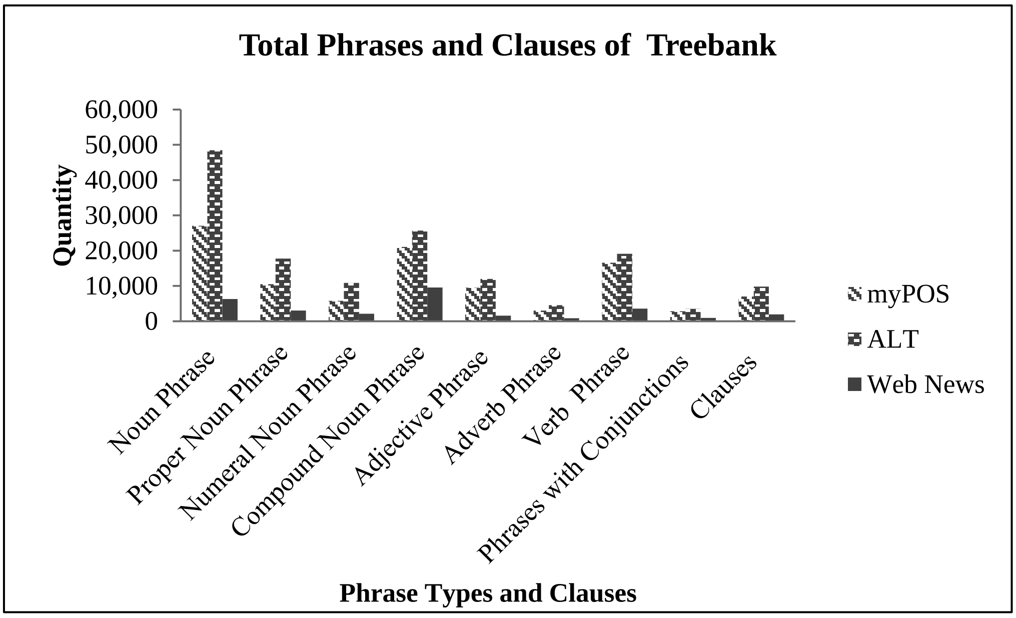


Figure 4.21 Phrases and clauses in Myanmar dependency treebank

4.8.1 Dependency Relations

Dependency relations are one main part of dependency corpus or treebank and can provide the role of the content word in sentence. Moreover, they are also useful for dependency parsing and other natural language processing application such as machine translation, question and answering, etc.

It can also be annotated by unsupervised or manual supervised way. Current unsupervised annotated dependency relations in each corpus of treebank are listed in

Table 4.24. Total dependency relation labels of current Myanmar treebank are shown in Figure 4.22.

Table 4.24 Frequencies of dependency labels in Treebank

Typed labels	Description	myPOS	ALT	Web News
case	Case Marking	71,890	108,177	16,601
obl	Oblique Noun	43,279	62,750	11,856
acl	Clausal Modifier of Noun	28,018	57,288	9,970
compound	Compound	26,898	39,583	9,917
mark	Marker	25,193	42,387	6,842
punct	Punctuation	16,100	27,499	4,144
advmod	Adverbal Modifier	7,488	11,065	2,447
nummod	Numeric modifier	6,022	11,251	2,255
amod	Adjectival Modifier	3,163	3,920	732
nmod	Nominal Modifier	454	54	146
root	Root	11,010	10,000	1,800
aux	Auxiliary	7	0	0
obj	Object	6	0	0
iobj	Indirect Object	3	0	0
nsubj	Nominal Subject	1	0	0
fixed	Fixed Multiword Expression	1	0	0
dep	Unspecified Dependency	1	0	0

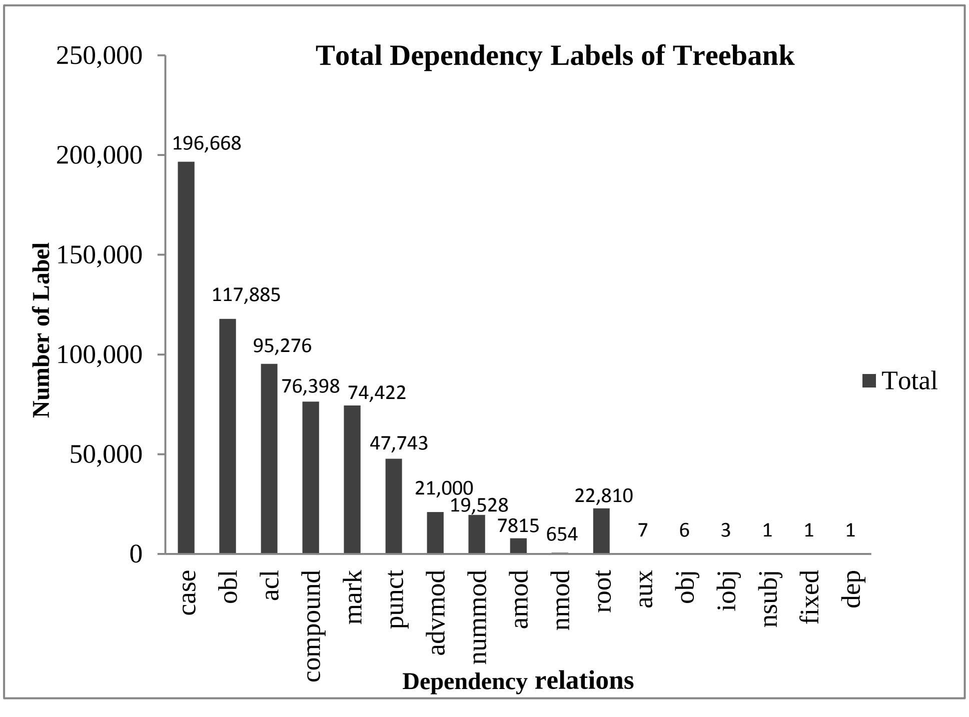


Figure 4.22 Total dependency labels in Myanmar dependency treebank

4.9 Chapter Summary

The procedures of building treebanks and the process of building Myanmar dependency treebank have been described. And challenges occurred in fundamental steps of building Myanmar treebank have been presented. Dependency annotation schemes for Myanmar treebank have been described with examples. Moreover, data statistics information of Myanmar dependency treebank such as number of sentences, range of the length of sentences, phrases, and clause structures, dependency relation labels have been analyzed and presented.

CHAPTER 5

UNSUPERVISED DEPENDENCY PARSING FOR MYANMAR LANGUAGE

Parsing Myanmar sentences is a critical part of natural language processing research area for Myanmar language. Among two main types of parsing, syntax or phrase parsing and dependency parsing, dependency parsing is simpler and better than phrase parsing to represent syntactic and semantic information for any language. Moreover, dependency parsing has become a prime focus of NLP research area due to having ability to help parsing free word order languages.

In dependency parsing, two CoNLL shared tasks [57] and [28] were important milestones ten years ago, and provided 20 treebanks of different languages with the same annotation format. Moreover, in unsupervised dependency parsing area, there has been a big progress to be able to parse raw input sentence without using any annotated treebank as the motivation. In the recent years, many researchers are working together on a project known as Universal Dependencies (UD) project which is a collection of treebanks for many languages, where the different morphological and dependency annotation styles of those languages are unified [13].

Being free word order language, unsupervised dependency parsing is proposed for Myanmar language parsing to have Myanmar language. In order to apply unsupervised parsing approach, CoNLL–U format and Universal part-of-speech tags have been applied as our first contribution for data representation format in Myanmar dependency parsing.

In unsupervised Myanmar dependency parsing, in order to build dependency parsing model, segmented and POS-tagged corpora data were annotated by UDPipe which is open source shared pipeline processing tool of multilingual parsing research project in which some languages have been joined and their Universal Dependencies (UD) format treebanks have been also shared for the development of multilingual parsing across languages. UDPipe can perform easily segmentation, POS tagging and parsing [39].

With the big progress of unsupervised dependency parsing field, unsupervised dependency parsing becomes only probable way to get syntactic information for under or low level resource languages. Being in still under resource stage for Myanmar syntactic dependency information, unsupervised dependency parsing approach is proposed for dependency parsing for Myanmar language as one contribution of this research motivation.

The statistical methods attained better quality compared to the human annotations rather than rule-based ones. Dependency parsing can be performed by two parsing types: grammar-driven and data-driven parsing. Among them, data-driven parsing is mostly applied for free word order language.

Two main approaches applied in data-driven parsing are using graph-based models and transition-based models. Graph-based models define a space of candidate dependency graphs for a sentence. And they induce a model for scoring an entire dependency graph for a sentence as learning processing for parsing. Final parsing is based on finding the highest-scoring dependency graph, given the induced model. It is difficult to define a space of candidate dependency graphs for learning process of input sentence to produce optimal parse tree with highest score in graph-based dependency parsing.

Transition-based models define a transition system for mapping a sentence to its dependency graph. Moreover, they induce a model for the next state transition prediction, given the transition history in transition configuration process to choose optimal transition for pars tree. Parsing is based on building the optimal transition sequence, given the induced model. UDPipe pipeline processing tool uses transition-based dependency parsing method based on transition predictions of neural network classifier.

Myanmar language has free word order nature and sentences can also be written in relatively free style. And one syntactic role of sentence can be one or more words or phrases or clauses in Myanmar sentences.

Therefore, in graph-based dependency parsing, it is difficult to define a particular space of candidate dependency graphs for learning process of input Myanmar sentence to produce optimal parse tree with highest score.

Therefore, dependency parsing with transition-based model is proposed for Myanmar dependency parsing.

5.1 Transaction-Based Dependency Parsing

Transition-based dependency parsing predicts a transition sequence from an initial configuration to some terminal configuration, which derives a target dependency parse tree, as shown in Figure 5.1. Greedy parsing uses a classifier for predicting the correct transition based on features extracted from the configuration. The greedy parsers are in a great interest because of their efficiency, although their performance is slightly worse than the search based parsers because of subsequent error propagation. The greedy parser is used in transaction-based dependency parsing.

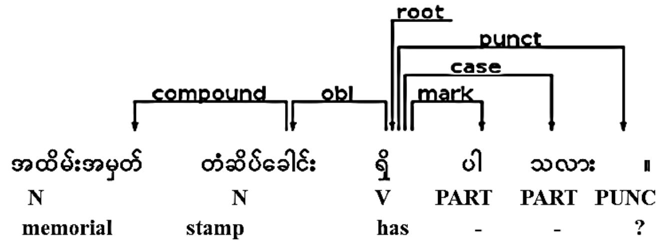


Figure 5.1 Example dependency tree

As the basis of transaction-based dependency parser, the arc-standard system [26], one of the most popular transition systems is employed. The arc standard system has a configuration ***c = (s, b, A)*** that consists of a stack ***s***, a buffer ***b***, and a set of dependency arcs ***A***.

The initial configuration for a sentence, $w_1,\ldots,w_n$ is s = [ROOT]; b = [$w_1,\ldots,w_n$]; $A = \emptyset$. A configuration c is terminal if the buffer is empty and the stack contains the single node ROOT, and the parse tree is given by A_c. Denoting s_i (i = 1,2,….) as the i^{th} top element on the stack, and b_i (i = 1,2,…) as the i^{th} element on the buffer, the arc-standard system defines three types of transitions:

LEFT-ARC(l): adds an arc s1 → s2 with label l and removes s2 from the stack. Precondition: $|s| \geq 2$.

RIGHT-ARC(l): adds an arc s2 → s1 with label l and removes s1 from the stack. Precondition: $|s| \geq 2$.

SHIFT: moves b1 from the buffer to the stack. Precondition: $|b| \geq 1$.

In the labeled version of parsing, there are in total $|T|= 2N_l + 1$ transitions, where N_l is number of different arc labels. Figure 5.2 illustrates an example of one transition sequence from the initial configuration to a terminal one. Figure 5.3 also illustrates an example intermediate configuration.

Configuration			
Transition	**Stack = s[…3,2,1]**	**Buffer = b[1,2,3,…..]**	**Configuration: A**
	[ROOT]	[အထိမ်းအမှတ် တံဆိပ်ခေါင်း ရှိပါ သလား။]	∅
SHIFT	[ROOT အထိမ်းအမှတ်]	[တံဆိပ်ခေါင်း ရှိပါ သလား။]	
SHIFT	[ROOT အထိမ်းအမှတ် တံဆိပ်ခေါင်း]	[ရှိ ပါ သလား။]	
LEFT -ARC (compound)	[ROOT တံဆိပ်ခေါင်း]	[ရှိ ပါ သလား ။]	A ∪ compound (တံဆိပ်ခေါင်း , အထိမ်းအမှတ်)
SHIFT	[ROOT တံဆိပ်ခေါင်း ရှိ]	[ပါ သလား ။]	
LEFT -ARC (obl)	[ROOT ရှိ]	[ပါ သလား။]	A ∪ obl (ရှိ , တံဆိပ်ခေါင်း)
SHIFT	[ROOT ရှိ ပါ]	[သလား။]	
RIGHT-ARC (mark)	[ROOT ရှိ]	[သလား။]	A ∪ mark (ရှိ , ပါ)
SHIFT	[ROOT ရှိ သလား]	[။]	
RIGHT-ARC (case)	[ROOT ရှိ]	[။]	A ∪ case (ရှိ , သလား)
…	…	…	…
RIGHT-ARC (root)	[ROOT]	[]	A ∪ root (ROOT, ရှိ)

Figure 5.2 Example transition sequence of the arc-standard system

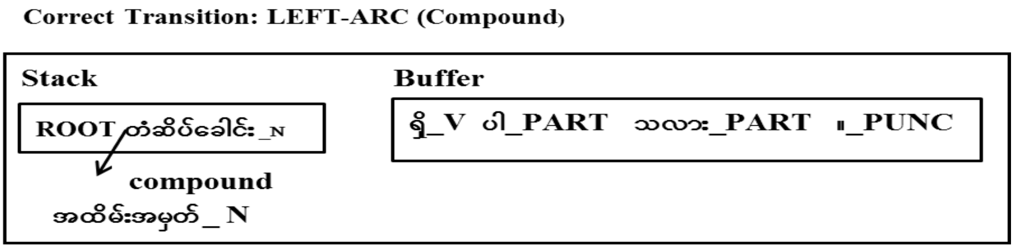


Figure 5.3 An intermediate configuration

The essential goal of a greedy parser is to predict a correct transition from *T*, based on one given configuration. Information that can be obtained from one configuration includes: (1) all the words and their corresponding POS tags (e.g., ရှိ / **V**); (2) the head of a word and its label (e.g., **obl, mark**) if applicable; (3) the position of a word on the stack/buffer or whether it has already been removed from the stack.

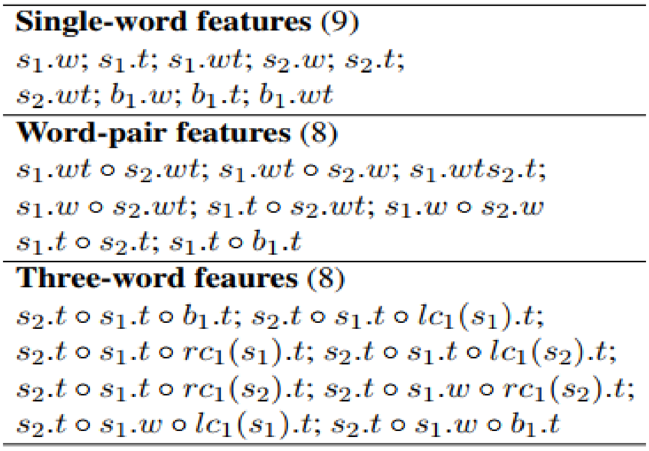
Conventional transition-based dependency parsing approaches extract indicator features such as the conjunction of 1 to 3 elements from the stack/buffer using their words, POS tags or arc labels. Table 5.1 lists a typical set of feature templates chosen from the ones of [36] [71]. In Table 5.1, $lc_1(s_i)$ and $rc_1(s_i)$ denote the leftmost and rightmost children of s_i, *w* denotes word, *t* denote POS tag. These features suffer from the following problems:

Sparsity. In many NLP tasks, the features, especially such as lexicalized features are highly sparse, and this is a common problem. In dependency parsing, the situation is severe because it depends critically on the interactions of word-to-word and thus the high-order features.

Incompleteness. An unavoidable issue in all existing feature templates is incompleteness. They still do not include the conjunction of every useful word combination although expertise and manual handling involved. For example, the conjunction of s_1 and b_2 is omitted in almost all commonly used feature templates, however it could indicate that if there is an arc from s_1 to b_2, a RIGHT-ARC action cannot be performed.

Expensive feature computation. The feature generation of indicator features is generally expensive — some words, POS tags, or arc labels have to be concatenated for generating feature strings, and look them up in a huge table containing several millions of features.

Table 5.1 The feature templates used for analysis



Single-word features (9)
$s_1.w$; $s_1.t$; $s_1.wt$; $s_2.w$; $s_2.t$; $s_2.wt$; $b_1.w$; $b_1.t$; $b_1.wt$
Word-pair features (8)
$s_1.wt \circ s_2.wt$; $s_1.wt \circ s_2.w$; $s_1.wts_2.t$; $s_1.w \circ s_2.wt$; $s_1.t \circ s_2.wt$; $s_1.w \circ s_2.w$ $s_1.t \circ s_2.t$; $s_1.t \circ b_1.t$
Three-word feaures (8)
$s_2.t \circ s_1.t \circ b_1.t$; $s_2.t \circ s_1.t \circ lc_1(s_1).t$; $s_2.t \circ s_1.t \circ rc_1(s_1).t$; $s_2.t \circ s_1.t \circ lc_1(s_2).t$; $s_2.t \circ s_1.t \circ rc_1(s_2).t$; $s_2.t \circ s_1.w \circ rc_1(s_2).t$; $s_2.t \circ s_1.w \circ lc_1(s_1).t$; $s_2.t \circ s_1.w \circ b_1.t$

The neural network model for learning dense features along with experimental evaluations was elaborated and has been proved its efficiency in [10]. Experimental evaluations of a transition-based dependency parser using neural networks outperformed other greedy parsers using sparse indicator features in both accuracy and speed by representing dense vectors with all words, POS tags, and arc labels. The model relied on dense features is able to automatically learn the most useful feature conjunctions for making predictions. Most Myanmar sentences have free word order nature and consist of long phrases or clauses. In order to apply better automatic learning the most useful features of Myanmar words for making predictions, transition-based dependency parsing with neural networks classifier is proposed for dependency parsing of Myanmar sentences.

5.1.1 Neural Network Classifier

The architecture of neural network classifier is illustrated in Figure 5.4. For usual word embeddings, each word as a d-dimensional vector is $\mathrm{e}_{\mathrm{i}}^{\mathrm{w}} \in \mathrm{R}^{\mathrm{d}}$and the full embedding matrix is $\mathrm{E}^{\mathrm{w}} \in \mathrm{R}^{\mathrm{d} \times \mathrm{N_w}}$ where $\mathrm{N_w}$ is the dictionary size. Meanwhile, POS tags and arc labels are mapped to a d-dimensional vector space, where $\mathrm{e}_{\mathrm{i}}^{\mathrm{t}}, \mathrm{e}_{\mathrm{j}}^{\mathrm{l}} \in \mathrm{R}^{\mathrm{d}}$ are the representations of i^{th} POS tag and j^{th} arc label. Respectively, the matrices of POS and label embedding are $\mathrm{E}^{\mathrm{w}} \in \mathrm{R}^{\mathrm{d} \times \mathrm{N_t}}$ and $\mathrm{E}^{\mathrm{l}} \in \mathrm{R}^{\mathrm{d} \times \mathrm{N_l}}$ where N_t and N_l are distinct POS tags and arc labels numbers. The model chooses a set of elements on the stack/ buffer positions for each type of information which might be word, POS or label, which might be useful for predictions denoted as the sets: S^w, S^t , S^l respectively [10]. Example configuration is illustrated in Figure 5.2 and St = $\{lc_1\ (s_2).\mathrm{t}, s_2.\mathrm{t}, rc_1\ (s_2).\mathrm{t}, s_1.\mathrm{t}\}$to extract "**N, N, V,..., PUNC**" in order.

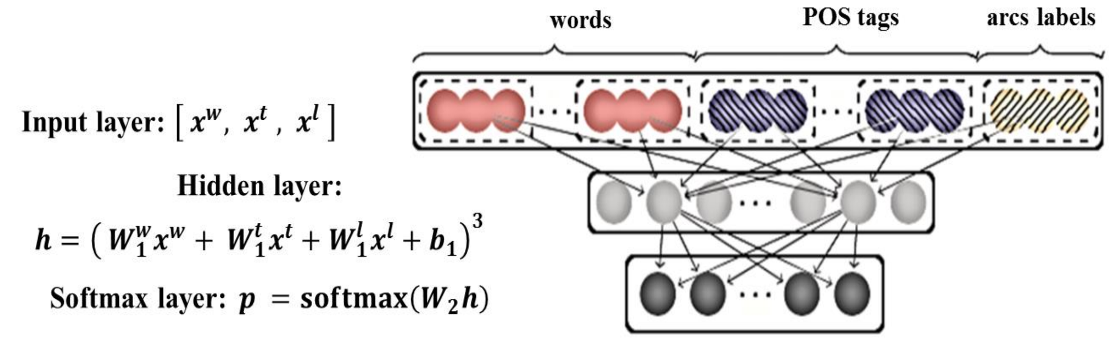

words
POS tags
arcs labels
Input layer: $[\, x^w,\ x^t\ ,\ x^l \,]$
Hidden layer:
$h = \left(W_1^w x^w + W_1^t x^t + W_1^l x^l + b_1 \right)^3$
Softmax layer: $p\ = \mathrm{softmax}(W_2 h)$

Figure 5.4 Neural network classifier architecture

A standard neural network is built with one hidden layer, where the corresponded embedding of the chosen elements from S^w, S^t , S^l will be added to the input layer. Denoting n_w , n_t , n_l as the chosen elements numbers of each type, x^w = [e^w_{w1}, e^w_{w2} ,$e^w_{wn_w}$] is added to the input layer, where $S^w = \{w_1, \dots, w_{n_w}\}$. Similarly, x^t and x^l are added to the input layer for the POS tag features and arc label features respectively. The input layer is mapped to a hidden layer with d_h nodes through a cube activation function:

$$h = (W_1^w x^w + W_1^t x^t + W_1^l x^l + b_1)^3 \qquad \textbf{(5.1)}$$

In (5.1), $W_1^w \in R^{d_h \times (d.\ n_w)}$,$W_1^t \in R^{d_h \times (d.\ n_t)}$,$W_1^l \in R^{d_h \times (d.\ n_l)}$, and $b_1 \in R^{d_h}$ is the bias. Finally, for modeling multi-class probabilities p = softmax (W_2h), where $W_2 \in R^{|T| \times d_h}$, a softmax layer is added on the top of the hidden layer [10] .

5.2 Corpus Preparation for Myanmar Dependency Parsing

Before implementing the Myanmar dependency parsing, dependency treebank data are needed. Therefore, the sentences from myPOS corpus, ALT parallel corpus, and Web News corpus were collected to build Myanmar dependency treebank. These corpora were created with different segmented and POS tagged styles. Therefore, it is important to check not only the word segmentation but also POS tag of each word token in sentences of these corpora to be required unique format of Myanmar treebank construction before the data preprocessing step.

For making unique formation, using only thorough manual word segmentation and POS tagging can provide more reliable tokenized sentences with POS tags for various written styles of Myanmar sentences.

Therefore, the manual method was used to check again tokenization and the collected sentences from the above three corpora with the new defined POS tagging scheme for Myanmar treebank construction in this research.

5.3 Myanmar Dependency Parsing

There are two main phases in Myanmar dependency parsing as an overview structure. First is training for building model. Second is parsing input test sentences with the model. In training part, data preprocessing will be carried out before building trained model. For proposed transition-based data driven parsing and for testing parsing performance, data training process with required format is need to build the model.

Data preprocessing is needed to be carried out to be required complete format for training data to build the model. Before the data preprocessing step, corpus preparation step is need to be carried out due to different corpora are used to build train model as described in the above Section. Data preprocessing stage consists of three steps: adding Universal-part-of-speech (U-POS) tags (with CoNLL-U format), unsupervised annotation to add dependency structures, and post processing unsupervised annotated results. After post processing, training with post processed data will be carried out to build the parse model.

Parsing input test sentences will be carried out after first training model phase. Input plain sentence will be passed in word segmentation and POS tagging step. Then, segmented and POS tagged sentence will be added to Universal part-of-speech tags in order to parse with trained model in next step. Finally, parsed tree with dependency structures will be generated as an output of parsing step. System architecture and work flow of processes can be viewed in Figure 5.5.

5.3.1 Data Preprocessing

As the Universal-part-of-speech tags and unsupervised dependency parsing approach are proposed, the training data must be the format of Universal Dependencies (UD). CoNLL-U format supports UD format. Therefore, training data must be CoNLL-U format in order to add U-POS and apply unsupervised annotation to get raw dependency syntactic structures for Myanmar language in faster way. It is also needed to post process on raw annotated dependency structures in order to parse new Myanmar sentences according to grammar structures of Myanmar language. The model trained with post processed data can provide more close dependency tree to Myanmar grammar structures rather than model trained with pure unsupervised annotated data which can generate easily a dependency tree.

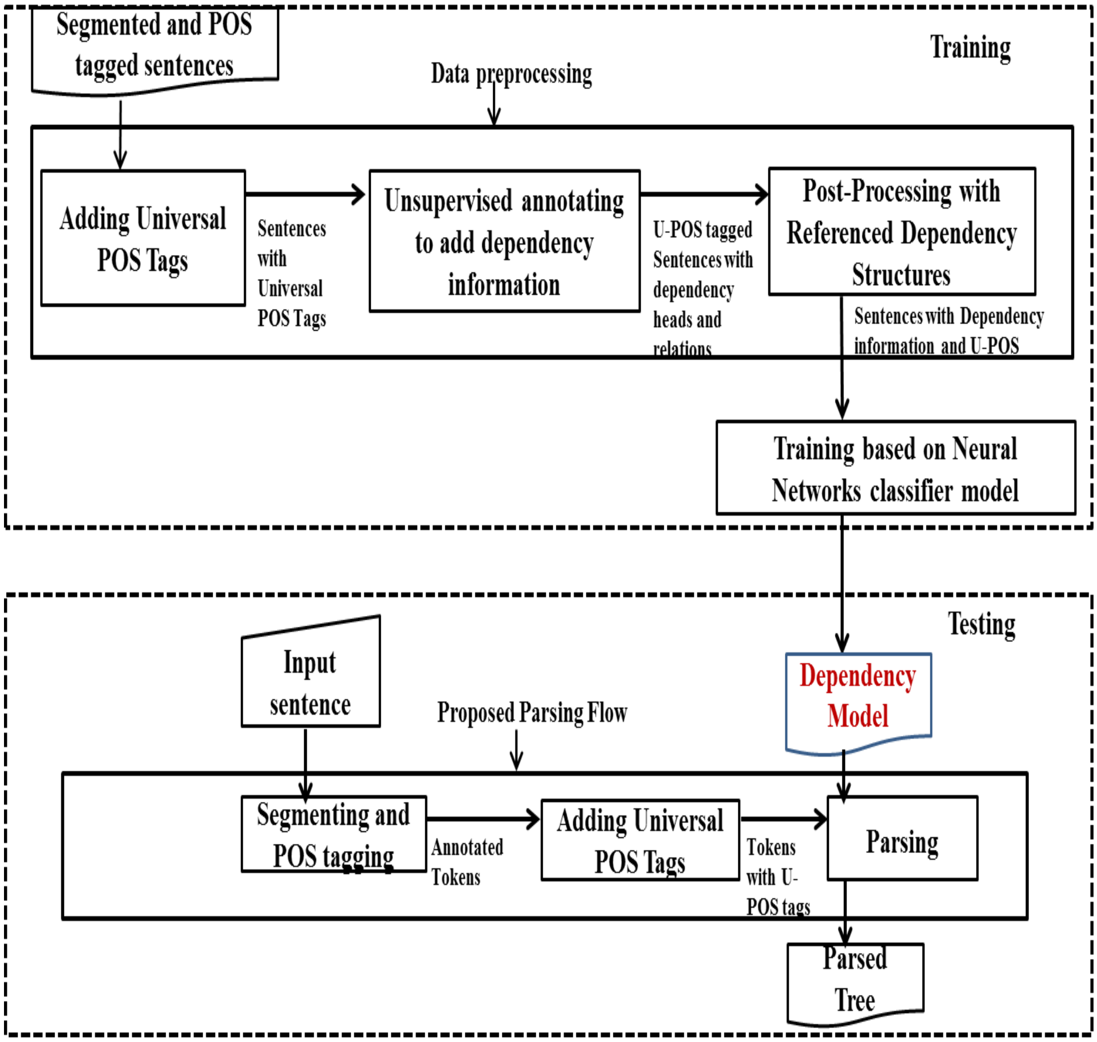

Segmented and POS tagged sentences
Training
Data preprocessing
Adding Universal POS Tags
Sentences with Universal POS Tags
Unsupervised annotating to add dependency information
U-POS tagged Sentences with dependency heads and relations
Post-Processing with Referenced Dependency Structures
Sentences with Dependency information and U-POS
Training based on Neural Networks classifier model
Testing
Input sentence
Proposed Parsing Flow
Dependency Model
Segmenting and POS tagging
Annotated Tokens
Adding Universal POS Tags
Tokens with U-POS tags
Parsing
Parsed Tree

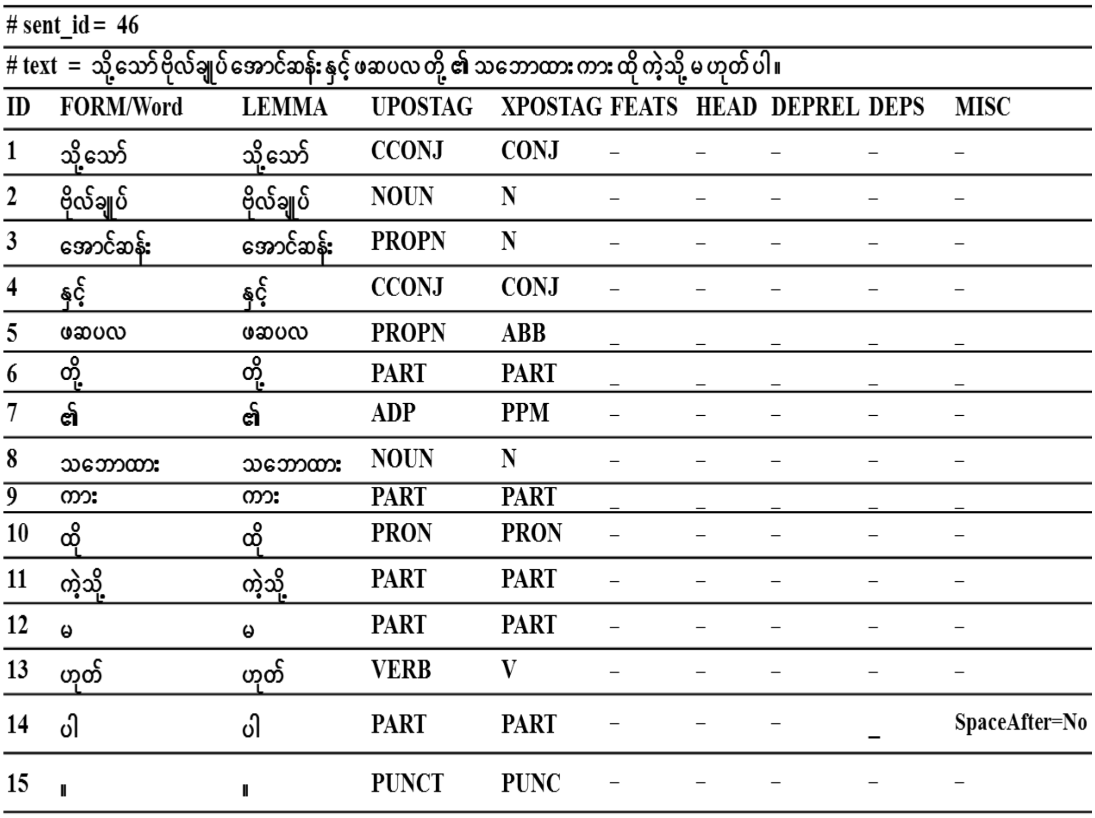
Figure 5.5 System flow diagram of Myanmar Dependency Parsing

5.3.1.1 Adding Universal Part-of-Speech Tags

In order to apply Universal Dependencies structure, training data must be converted to CoNLL-U format as mentioned in UD project to add related Universal-Part-of-Speech (U-POS) and specific language POS tags for words of sentence. U-POS and language POS tags are added in UPOSTAG and XPOSTAG column in CoNLL-U format.

In converting CoNLL-U format, there is a mapping process between U-POS tags and Myanmar language as the mapping scheme described in Section 4.4.1 of previous chapter. Most Myanmar language POS tags are very close to U-POS tags. Example sentence with CoNLL-U format is shown in Table 5.2.

Table 5.2 Sample Myanmar sentence with CoNLL-U format of train data



# sent_id= 46									
# text = သို့သော် ဗိုလ်ချုပ် အောင်ဆန်း နှင့် ဖဆပလ တို့ ၏ သဘောထား ကား ထို ကဲ့သို့ မ ဟုတ် ပါ။									
ID	FORM/Word	LEMMA	UPOSTAG	XPOSTAG	FEATS	HEAD	DEPREL	DEPS	MISC
1	သို့သော်	သို့သော်	CCONJ	CONJ	–	–	–	–	–
2	ဗိုလ်ချုပ်	ဗိုလ်ချုပ်	NOUN	N	–	–	–	–	–
3	အောင်ဆန်း	အောင်ဆန်း	PROPN	N	–	–	–	–	–
4	နှင့်	နှင့်	CCONJ	CONJ	–	–	–	–	–
5	ဖဆပလ	ဖဆပလ	PROPN	ABB	_	_	_	_	_
6	တို့	တို့	PART	PART	_	_	_	_	_
7	၏	၏	ADP	PPM	–	–	–	–	–
8	သဘောထား	သဘောထား	NOUN	N	–	–	–	–	–
9	ကား	ကား	PART	PART	_	_	_	_	_
10	ထို	ထို	PRON	PRON	–	–	–	–	–
11	ကဲ့သို့	ကဲ့သို့	PART	PART	–	–	–	–	–
12	မ	မ	PART	PART	–	–	–	–	–
13	ဟုတ်	ဟုတ်	VERB	V	–	–	–	–	–
14	ပါ	ပါ	PART	PART	–	–	–	_	SpaceAfter=No
15	။	။	PUNCT	PUNC	–	–	–	–	–

5.3.1.2 Unsupervised Annotation

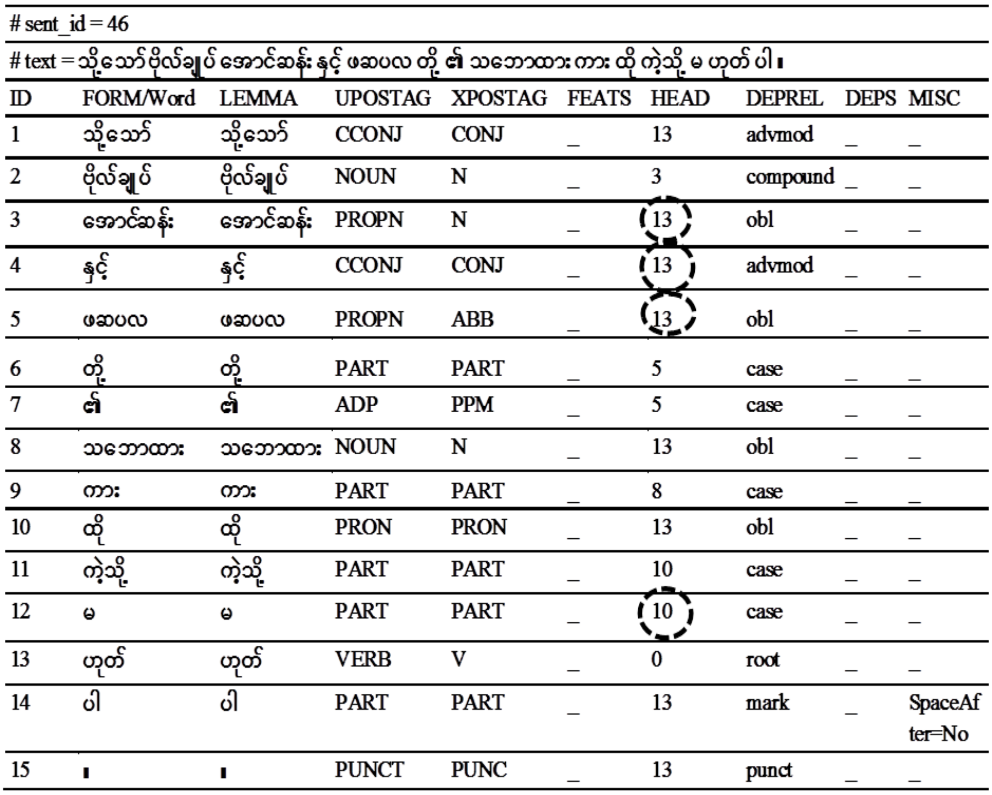
Universal Dependencies is a project developing cross-linguistically consistent annotation for many languages, with the purpose of multilingual parser development by cross-lingual learning, and parsing research from a language typology perspective by universal dependencies and shares tools and dependency models of many language treebanks. Therefore, applying unsupervised dependency parsing becomes a probable way to acquire dependency syntactic information for low or under resource languages such as Myanmar language. To apply unsupervised induction of dependency syntactic structure, related U-POS tag and universal dependency relation (UDEPL) feature are needed [p2].

Since any dependency syntactic information resource for Myanmar language had not been yet to annotated data for training data, unsupervised dependency parsing approach is used to annotate train data for parsing model building to get initial raw dependency information for Myanmar training data to reduce annotation time. Unsupervised corpus annotation was done by UDPipe pipeline process tool by using shared model of Japanese Universal treebank of UD project since grammar structures of Myanmar and Japanese languages are very similar. .

Since any dependency syntactic information resource for Myanmar language had not been yet to annotated data for training data, unsupervised dependency parsing approach is used to annotate train data for parsing model building to get initial raw dependency information for Myanmar training data to reduce time-consuming. Unsupervised corpus annotation was done by UDPipe pipeline process tool by using shared model of Japanese Universal treebank of UD project since grammar structures of Myanmar and Japanese languages are very similar.

As a significant success of unsupervised annotation, shared Japanese parsing model can provide rooted dependency trees at final main verbs of most sentences in corpus. Unsupervised annotated results by Japanese model can be viewed in Table 5.3.

Table 5.3 Example unsupervised annotated result



sent_id = 46

text = သို့သော်ပိုလ်ချုပ် အောင်ဆန်း နှင့် ဖဆပလ တို့ ၏ သဘောထား ကား ထို ကဲ့သို့ မ ဟုတ် ပါ ။

ID	FORM/Word	LEMMA	UPOSTAG	XPOSTAG	FEATS	HEAD	DEPREL	DEPS	MISC
1	သို့သော်	သို့သော်	CCONJ	CONJ	_	13	advmod	_	_
2	ပိုလ်ချုပ်	ပိုလ်ချုပ်	NOUN	N	_	3	compound	_	_
3	အောင်ဆန်း	အောင်ဆန်း	PROPN	N	_	13	obl	_	_
4	နှင့်	နှင့်	CCONJ	CONJ	_	13	advmod	_	_
5	ဖဆပလ	ဖဆပလ	PROPN	ABB	_	13	obl	_	_
6	တို့	တို့	PART	PART	_	5	case	_	_
7	၏	၏	ADP	PPM	_	5	case	_	_
8	သဘောထား	သဘောထား	NOUN	N	_	13	obl	_	_
9	ကား	ကား	PART	PART	_	8	case	_	_
10	ထို	ထို	PRON	PRON	_	13	obl	_	_
11	ကဲ့သို့	ကဲ့သို့	PART	PART	_	10	case	_	_
12	မ	မ	PART	PART	_	10	case	_	_
13	ဟုတ်	ဟုတ်	VERB	V	_	0	root	_	_
14	ပါ	ပါ	PART	PART	_	13	mark	_	SpaceAfter=No
15	။	။	PUNCT	PUNC	_	13	punct	_	_

In Table 5.3, all other dependency head values are correct except head values in circles. Dependency head values of words in circles are needed to update in manual post processing.

5.3.1.3 Post Processing

Unsupervised annotated dependency structures are correct in generally for the whole word orders dependency structures and main rooted of the whole sentence. However, some of the word dependency structures are needed to update according to the dependency structures of Myanmar grammar to build reliable parsing model. Before post processing step, dependency schemes are needed to define in order to be able to use as referenced dependency schemes as mentioned in Section 4.5.1 of previous chapter for post processing. Post processing was carried out manually.

In post processing, post processed data were trained again to build updated training model to annotate data which have not been post processed by UDPipe repeatedly to reduce errors in manual correction and to be more consistent format in manual post processing as a bootstrapping manner.

5.3.1.4 Training Model

Post processed training data are trained again by UDPipe using transitioin-based dependency parsing based on neural network classifier to build reliable Myanmar dependency parsing model for parsing new Myanmar sentences. Training the model after post processing generates the more reliable dependency model for dependency parsing of Myanmar sentences.

5.4 Parsing with Myanmar Trained Model

Parsing new Myanmar sentences will be carried out to test performance of trained Myanmar dependency parsing model. For parsing new sentences will be carried out by three main steps: segmenting and POS tagging process for input sentence, adding U-POS to CoNLL-U format and parsing by trained model.

5.4.1 Segmenting and POS Tagging on input

As a fundamental step of parsing process, input sentence is needed to be segmented. The segmented sentence will be tagged by Myanmar language specific POS tags, after that tokenized and POS tagged tokens of input sentence will be ready to convert CoNLL-U format.

5.4.2 Adding U-POS to CoNLL-U Format

In order to parse new Myanmar sentence by Universal Dependencies parsing model, it is needed to convert CoNLL-U format and to add U-POS by mapping scheme between Myanmar language specific POS tags and U-POS tags. Input word token sequence with CoNLL-U format with Myanmar language specific POS tags and U-POS tags will be ready to parse by proposed model.

5.4.3 Parsing Input Sentence

The converted input tokenized word sequence POS tagged by language POS tags and u-POS tags will be parsed by the proposed trained model based on transition predictions of neural network classifier.

5.5 Chapter Summary

Since parsing Myanmar sentences is still a critical part of Myanmar natural language processing research area, unsupervised dependency parsing has been proposed for Myanmar language. It uses the transition-based dependency parsing which chooses the best optimal transitions from the transaction predictions by neural networks classifier. Then building process of the parsing model for Myanmar language has been presented. Finally, how to parse input sentence to test the Myanmar parsing model have also been described in this chapter.

CHAPTER 6

EXPERIMENTAL RESULTS AND EVALUATIONS

This chapter will present two experiments relating with segmentation and POS tagging for the purpose of initial corpus preparation and three experiments inducted into direct dependency parsing for Myanmar sentences during this research. In addition, the detailed discussion on results and evaluation analysis will also be described.

First, two experiments were implemented for the word segment and POS tagged annotation performance of the corpus used in Asian Language Treebank (ALT) project.

Three experiments were implemented for Myanmar dependency parsing. The first one is the preliminary experiment for unsupervised dependency parsing on raw segmented and part-of-speech (POS) tagged corpus of Myanmar language. Moreover, the second is a general domain corpus annotation by an unsupervised approach via universal part-of-speech (U-POS) to construct Myanmar dependency treebank since manual annotation for complete structures of syntactic information of Myanmar sentences is a still challenging task. Word dependency structures in Myanmar sentences were also introduced as the overall dependency schemes for orders of phrases and words.

The third is the experimental result for the performance of proposed Myanmar dependency parsing model.

The evaluation and analysis of experiments and discussions will be presented by two main sections. As the first section, briefly description of experiments with methodologies and corpus data sets used in the experiments will be presented. And discussion and analyzing on the experimental results will be presented as the second part.

6.1 Experiments on ALT corpus Segmentation and Tagging Scheme

In order to know how part-of-speech (POS) information can support machine translation, corpus segmentation and tagging had been executed on ALT corpus as the first experiment. The overview of the first experiment is English-to-Myanmar statistical machine translation (SMT) based by using a language model on part-of-speech (POS) in decoding process since linguistic information for POS is valuable for statistical machine translation systems to extract translation rules. The second experiment is a study of comprehensive works in two main morphological analysis tasks: word segmentation and part-of-speech (POS) tagging, in Myanmar sentences.

6.1.1 Experiments in Statistical Machine Translation

Word segmentation and POS tagging scheme of ALT parallel corpus for Myanmar language was prepared for ALT treebank. An annotation system, called "NOVA", had been proposed for the joint word segmentation and POS tagging to use further modification and combining with adaptable complex linguistic phenomena [3]. It was used to annotate Myanmar corpus containing twenty thousand sentences of ALT corpus. In order to know how POS information can support to SMT and affect the translation performance of SMT, three experiments for English-to-Myanmar statistical machine translation were implemented by applying POS tagging scheme of Myanmar ALT parallel corpus and had been reported in [p1]. The word segmenting and POS tagging model of three SMT experiments was created by the ALT parallel corpus annotated by NOVA tag scheme. In Table 6.1, the statistic of ALT parallel Myanmar corpus is described.

Table 6. 1 Statistics on ALT parallel Myanmar corpus

Sentences	Tokens		
	Syllable	Short	Long
159,603	1,170,922	737,137	553,948

In English-to-Myanmar SMT experiments, the NOVA POS tag scheme was also used to get the POS information of train and test data of SMTs. Basic Travel Expression Corpus (BTEC) was used for train and test data set without specifying short and long tokens and statistics of these data set will be shown in Table 6.2.

Table 6.2 Statistics on BTEC corpus data set of English-to-Myanmar SMTs

Data set	#Sentences	#Tokens	
		English	Myanmar
Training	159,603	948,611	625,086
Development	1,622	9,748	6,413
Test	973	5,791	3,836

For the baseline system, PB SMT system of Moses toolkit was used [49]. For word alignment between source and target language, GIZA++ [17] is used and it was symmetrized by grow-diag-final-and heustristics [48]. The lexical reordering model was trained by the msd-bidirectional-fe option [8]. For the setting value of maximum phrase length is 9. To train the 9-gram language models (LM) by the way of interpolated modified Kneser-Ney discounting [58], SRILM [1] was used. For decoder setting, default parameters setting of Moses decoder except using distortion-limit value 12 were used to occur sufficient support from the LM. Two language models: untagged (LM) and POS tagged (POS LM) were used for three SMT experiments shown in first column one of Table 6.3.

Table 6.3 Result scores of the SMT systems

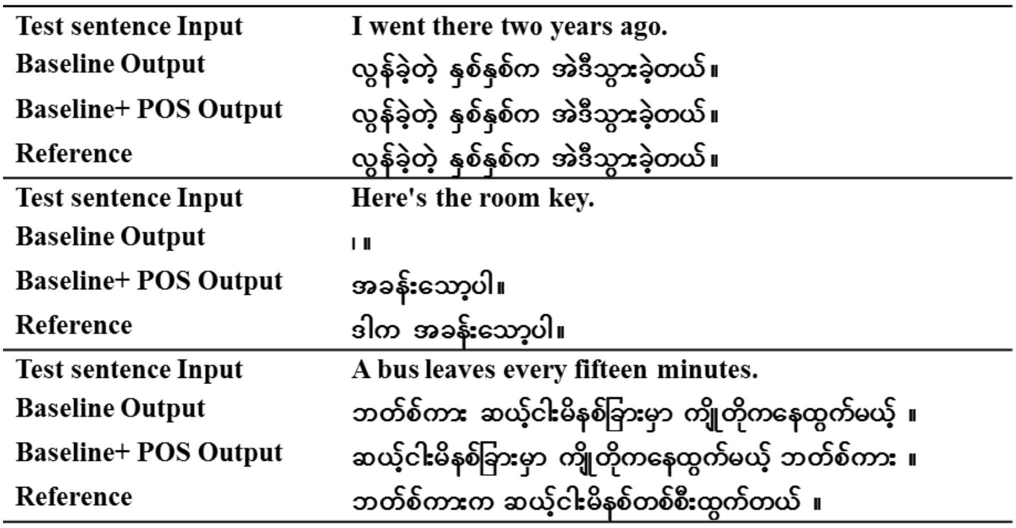
SMT systems	BLEU	RIBES
Baseline + LM	40.9	.564
Baseline + POS + LM	40.9	.564
Baseline + POS+ POS LM	41.0	.574

6.1.1.1 Results and Discussion of SMT Experiments

In SMT systems of Table 6.2, the first one is baseline SMT with language model (LM), the second one is baseline SMT which was added POS to target language training and used LM, and the third one is baseline SMT which was added POS to target language training and used POS LM.

To tune parameters of the decoder, minimum error rate tuning (MERT) was used. The translation results of test data set were evaluated by two automatic measurement scores: bilingual evaluation understudy (BLEU) which was used to measure the adequacy of the translations and rank-based intuitive bilingual evaluation measures (RIBES) which will penalize the wrong word orders. The higher BLEU score and the larger RIBES indicate the better performance.

The result of BLEU and RIBES of baseline SMT which used POS on LM and train data was a little improvement on first and second SMT systems of table 6.2. However, baseline PB SMT with POS SMT systems, second and third SMT of table 6.2, could produce 468 outputs as the reference ones while the baseline produced 461 outputs for 973 test sentences. Comparing different outputs of SMT systems with the reference sentences, PB SMT systems used POS information could produce better and more meaningful results rather than the baseline system as output sentences of the second and third input test sentences shown in Figure 6.1 although they are not fully correct as references.



Test sentence Input	**I went there two years ago.**
Baseline Output	လွန်ခဲ့တဲ့ နှစ်နှစ်က အဲဒီသွားခဲ့တယ်။
Baseline+ POS Output	လွန်ခဲ့တဲ့ နှစ်နှစ်က အဲဒီသွားခဲ့တယ်။
Reference	လွန်ခဲ့တဲ့ နှစ်နှစ်က အဲဒီသွားခဲ့တယ်။
Test sentence Input	**Here's the room key.**
Baseline Output	၊ ။
Baseline+ POS Output	အခန်းသော့ပါ။
Reference	ဒါက အခန်းသော့ပါ။
Test sentence Input	**A bus leaves every fifteen minutes.**
Baseline Output	ဘတ်စ်ကား ဆယ့်ငါးမိနစ်ခြားမှာ ကျိုတိုကနေထွက်မယ့် ။
Baseline+ POS Output	ဆယ့်ငါးမိနစ်ခြားမှာ ကျိုတိုကနေထွက်မယ့် ဘတ်စ်ကား ။
Reference	ဘတ်စ်ကားက ဆယ့်ငါးမိနစ်တစ်စီးထွက်တယ် ။

Figure 6.1 Sample translation outputs and references

As a conclusion on those experiments, they could be proved that SMT experiments using POS information are able to improve the translation performance even though the results of evaluation metrics were not much different in the three experiments at that moment.

6.1.2 Experiments in Morphological Analysis

Being typically strong head-finalization at syntax in Myanmar sentence, the functional dependent morphemes succeed morphemes of their main independent content. Moreover, the main root verb constituent of Myanmar sentences always occurs at the end of sentences. Subordinate clauses always modify their following parts which might be words, or phrases, or main clauses in sentences.

A study of comprehensive work in two main morphological analysis tasks in Myanmar sentences: word segmentation and POS-tagging in Myanmar textual data released by ALT project, has been conducted in [p4] being the above syntactic natures of Myanmar sentences. That study contributes to both practical works of NLP and linguistics area.

Firstly, Myanmar corpus was constructed linguistically by the way of two-layer word tokenized and POS-tagged scheme, called “**nova**”, to provide morphological information. It contains 20,000 sentences. The annotation scheme was also considered to cover important and basic linguistic phenomena of Myanmar sentences.

The "**nova**" annotation scheme includes four main tags, "**n"** represents noun, "**v"** represents verbs, "**a"** represents adjectives, and "**o"** represents other modification tokens, and three additional auxiliary tags: "**1"** which represents numbers, "**."** which represents punctuation marks, and "+" which represents weak syntactic roles tokens. In order to address the functionalities of three basic tags, a "-" mark can be additional modified as the three following tags: "**n-**" which represents **various pronouns** : personal, demonstrative, numeral ones, and interrogative**, "a-"** which represents **determiners** most derived from "**n-**", and "**o-**" which represents **other functional token words**: particles, positional markers and conjunctions. The basic tag **v** is not usually modified as **v**-. In addition, **a bracket pair form**, **"([and])"**, can be used to provide the two-layer annotation scheme to tag as an integrated unit which represent the large syntactic constituents for further syntactic parsing which is adaptable to the ambiguities most occurred in segmentation. Figure 6.2 shows an annotation example of Myanmar sentence.

Myanmar ALT corpus was cross-checked repeatedly seven times to be consistent and precise annotation. In order to increase the annotation quality in cross-checking time, the repeated manual refinement and automatic cross-validation tests were applied.

Table 6.4 Data setup for train and test of ALT corpus

Data Sets	**Syllable Tokens**	**Word Tokens**		**Sentences**
		Short	**Long**	
Training	1,054,829	664,174	498,227	17,965
Development	57,607	36,133	27,081	993
Test	58,486	36,830	27,740	1,007
Total	1,170,922	737,137	553,048	19,965
Average syllable(s)	1	1.59	2.12	58.65

In Table 6.4, the statistics of Myanmar ALT corpus used in morphological analysis experiments are listed with the token number of syllables, short word, and long words bracketed and assumed as long tokens.

The experiments of tokenization and POS-tagging of Myanmar sentences have been executed by two approaches: the approach of the conditional random fields

(CRFs) for standard sequence-labeling [30], and the recurred neural network (RNN) [31] with short-term memory (LSTM) [59].

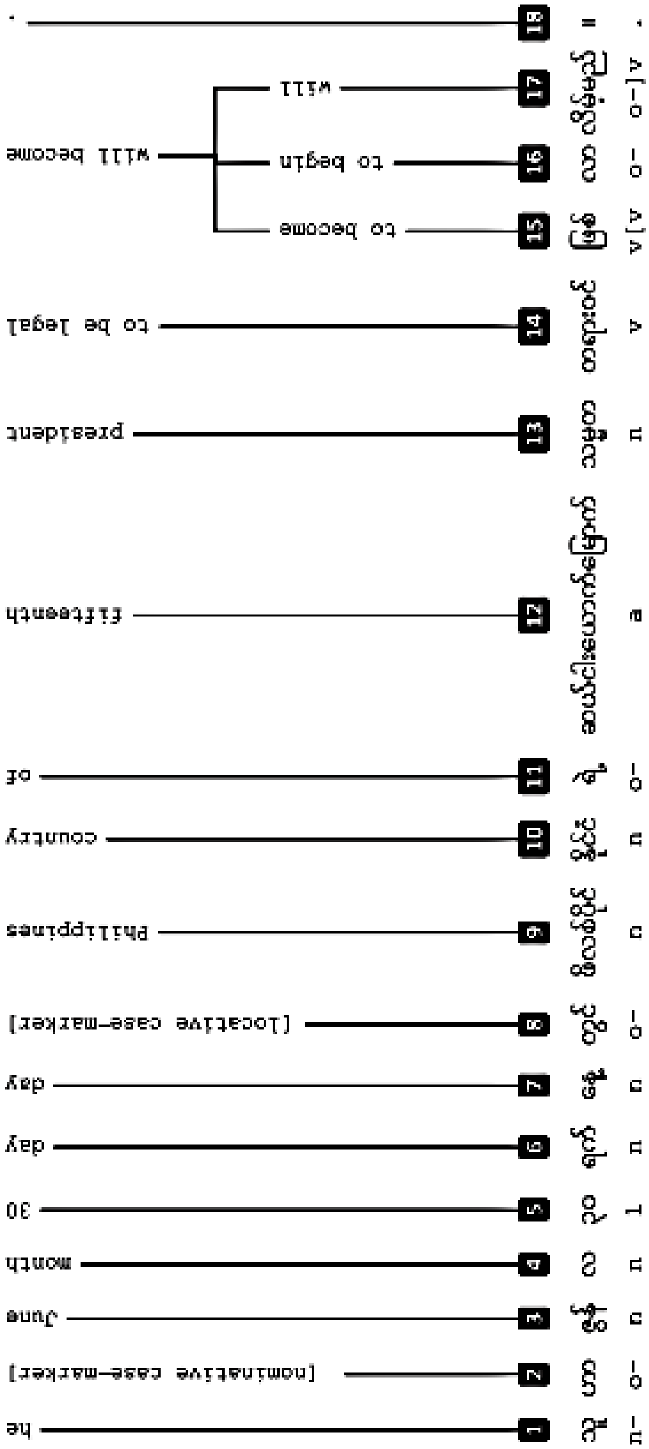


Figure 6.2 A sample tokenized and POS-tagged sentence.

6.1.2.1 Experiments By CRF

The fundamental components of the morphological analysis in Myanmar sentences are syllables and it is reasonable for the terms of linguistic structures and NLP works of Myanmar language. Since the basic unit is syllable, annotation scheme called IBES has been adopted. The four tags, **BEIS,** represent the **beginning** of a token with **B**, the **end** of a token **E**, the inside token with **I**, and the **single** unit with **S**, respectively. Only **IBES**, **IE**, and **IB** schemes are used to compare in morphological analytical experiments. In Figure 6.3, an example tagging scheme of IBES of the word tokenization of the last fragment in sample sentence is shown. The tags of IBES scheme can also be more attached with other POS tags for joint word segmentation and POS-tagging.

REL. IDX.:	-5	-4	-3	-2	-1	0	1	2	3	4	5
SYLLABLE:	သ	မ္မ	တ	တ	ရား	ဝင်	ဖြစ်	လာ	လိမ့်	မည်	။
SMALL-TOK:	**B**	**I**	**E**	**B**	**I**	**E**	S	S	**B**	**E**	S
LARGE-TOK:	**B**	**I**	**E**	**B**	I	**E**	**B**	**I**	**I**	**E**	S
SMALL-POS:	**B**-n	**I**-n	**E**-n	**B**-v	**I**-v	**E**-v	**S**-v	**S**-o-	**B-o-**	**E-o-**	**S**-.
LARGE-POS:	**B**-n	**I**-n	**E**-n	**B**-v	**I**-v	**E**-v	**B**-v	**I**-v	**I**-v	**E**-v	**S**-.

Figure 6.3 IBES tagging scheme for tokens 13, 14, 15, 16, 17, 18 in Figure 6.2

It turns into an **IE** scheme by changing **B** to **I** and **S** to **E**; it turns into an **IB** scheme by changing **E** to **I** and **S** to **B**. For an example case of stacked consonant letters, a stacked consonant letter kept together with main consonant letter, “္မ”, can be seen at the position number “**-4**”.

Basing on the syllables, the three feature templates were used for the morphological experiments, where S_m^n stands for the syllable sequence of the relative indices within [m, n]. And three templates are used for 1-gram, 2-gram, 3-gram, and 4-gram templates for context window sizes, 2, 3 and 4. CRF++ toolkit was used in the experiments. Specifically, the automatic segmented tokens of a Myanmar sentence are compared with the tokens manually segmented in testing. Then, the F-score was calculated in terms of tokens as the average precision and recall.

6.1.2.2 Results and Discussion of Experiments by CRF

The main results of the ALT short test tokens processing by the data of fully training with different combinations of feature-tag were summarized in Table 6.5. In Table 6.5, separate tokenization column shows the plain tokened results without using POS tags.

The joint tokenized and POS-tagged result columns include the simultaneous generating word tokens and POS tags results combining with extra POS tags to form a larger tag set. Indeed, the F-scores of word tokens without POS tagging and joint word tokens and POS tags as **(token/POS)** are also listed respectively. The compared results of separate tokens and the F-score of tokens from the process of the joint word segmentation and POS-tagging were also described as (Δjoint).

Table 6.5 Performance of CRFs in ALT short word tokens processing

Feature-tag	Separate tokenization		Joint tokenization and POS-tagging			
	accuracy	F-score	accuracy	F-score		
	(%)	token	(%)	token	(Δ_{joint})	token/POS
2-IBES	94.9	0.943	93.9	0.947	(+0.004)	0.940
3-IBES	95.1	0.944	93.9	0.946	(+0.002)	0.940
4-IBES	94.9	0.943	93.7	0.945	(+0.002)	0.938
2-IE	96.3	0.922	94.9	0.938	(+0.016)	0.931
3-IE	96.9	0.935	95.3	0.943	(+0.008)	0.937
4-IE	96.9	0.935	95.2	0.943	(+0.008)	0.936
2-IB	96.7	0.929	94.8	0.939	(+0.010)	0.932
3-IB	97.0	0.937	95.0	0.942	(+0.005)	0.935
4-IB	97.0	0.937	95.0	0.942	(+0.005)	0.936

In Table 6.5, as an obvious fact, the annotation schemes of **IE** and **IB** have much higher accuracies than the scheme of **IBES**, while it's F-score is noticeably lower. The annotation scheme of **IBES** codes more useful information, though it makes the more difficult task. Although appearing incredible tagging sequences may occur in the scheme of **IBES**, these contradictions seem rarely as one or two times or only with small amount times of training data like 1/16 and 1/8. Therefore, they are unimportant. The **IB** scheme is slightly better than the scheme of **IE** for the results of separate tokens as a minor point.

As the separate word tokenization process performance is already lower than the joint processed work, a two steps processing, first-tokenizing-then-POS-tagging, cannot accomplish better performance than the joint processed method.

For the setting of features on syllables, bi-grams (2-) seem adequate, and tri-grams (3-) bring restricted gain, but additional features (4-) will lead to performance degradation owing to sparseness and over-fitting.

Table 6.6 Performance of CRFs in ALT long tokens processing

Feature-tag	Separate tokenization		Joint tokenization and POS-tagging			
	accuracy	F-score	accuracy	F-score		
	(%)	token	(%)	token	(Δ_{joint})	token/POS
2-IBES	95.5	0.943	93.9	0.947	(+0.005)	0.929
3-IBES	95.6	0.933	94.3	0.936	(+0.003)	0.929
4-IBES	95.4	0.930	94.2	0.934	(+0.004)	0.927
2-IE	96.4	0.897	95.1	0.921	(+0.024)	0.913
3-IE	97.3	0.922	95.5	0.931	(+0.009)	0.924
4-IE	97.2	0.921	95.6	0.930	(+0.009)	0.923
2-IB	97.0	0.914	95.1	0.927	(+0.013)	0.919
3-IB	97.4	0.925	95.3	0.930	(+0.005)	0.923
4-IB	97.4	0.925	95.4	0.931	(+0.006)	0.924

Table 6.6 contains the results of long tokens of the ALT data. Specifically, the numerical results of long tokens are lower than short tokens of ALT. Therefore, bi-gram features are suitable for covering the short tokens' range of but comparatively unsatisfactory for long ones. However, the high-order features still do not obtain increased performance due to data sparseness. Subsequently, the long tokens processing becomes more challenging than that of short tokens.

6.1.2.3 Experiments By LSTM-RNN approach

The overall network structure of LSTM-RNN approach is illustrated in Figure 6.4, where a standard configuration setting for structured learning approach applied in several NLP tasks is encompassed by three modules: representation, classification, and structured interface. The component of representation is an automatic feature extractor, where discrete syllable tokens are encoded into an R^n space as an effectual representation.

Then the extracted features are fed into a fully related feed-forward neural network plain, which executes as a non-linear classifier. In this module, the non-linear transformation may be abbreviated [4] where all non-linear transformation was completed by LSTM units.

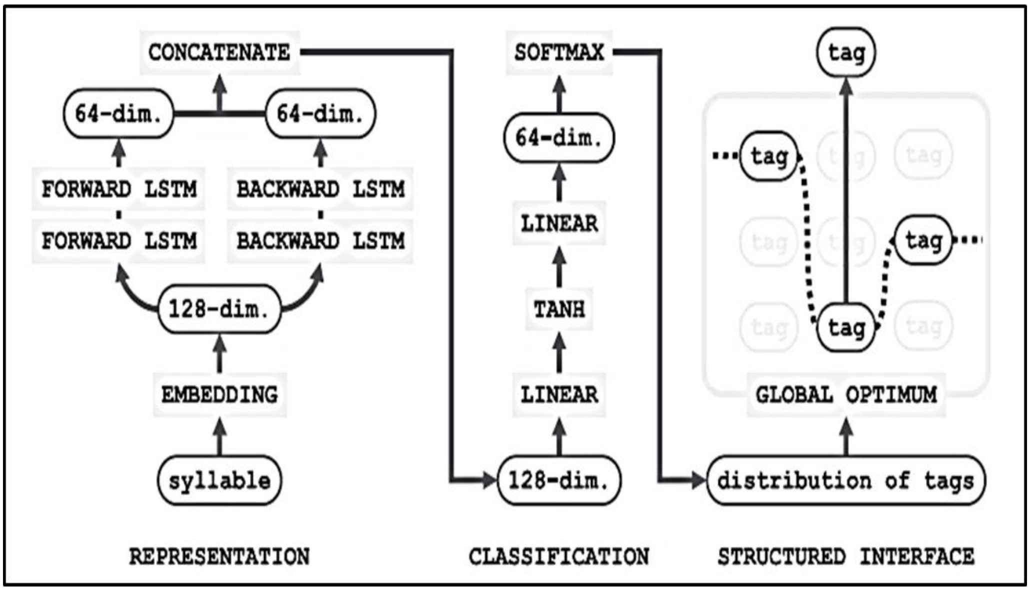


Figure 6.4 Configuration of the LSTM-RNN in morphological analysis

In the structured interface, only a standard Viterbi algorithm was applied to model for the relationship between neighboring tags, which is adequate for the training task. The two-layer bidirectional LSTM offering a trade-off between performance and speed of training process was used. A compact variant of LSTM with peepholes [15] was applied. The most common function**, tanh,** was applied. A practical ensemble method was discovered to combine large independently fast-trained networks to assuage the instability in performance generated by the process of initializing. The method is also robust and efficient from the perspective of improving the performance attained by single networks. A straightforward solution was applied to collect all possible bi-grams of output tags from the data of training and only possible selected sequences in the ensemble.

The version 2.0 DyNet toolkit [20] was used in the implementation. The model parameters are initialized by Xavier initializing method [69] and learned by Adam [12] in the experiments, after trying several parameter-optimizations that did not have much different performance. Dropout [47] value was not used, because it decelerated the training and this effect was not as important as that brought by the model ensemble. Table 6.7 summarizes the hyper-parameters setting for experiments.

Table 6.7 Hyper-parameters of LSTM-RNN

hyper parameter	value	Note
Dimension of embedding	128	Shown in Figure 6.4
Dimension of forward / backward LSTM states	64/64	
Dimension of feed-forward classification output	64	
Adam's learning rate ∝	$\mathbf{10^{-3}}$	Default by DyNet
Adam's moving average $\boldsymbol{\beta_1}$ for the mean	0.9	
Adam's moving average $\boldsymbol{\beta_2}$ for the variance	0.999	
Adam's bias ϵ	$\mathbf{10^{-8}}$	
Dropout rate	_	Disabled
Ensemble size	5,10,20,50, and 100	

6.1.2.4 Results and Discussion of Experiments by LSTM-RNN

The IBES tagging scheme was used consistently in all LSTM-RNN experiments, because it is the most efficient scheme, as discussed in above. The model numbers in the ensemble were also compared in order to investigate the ensemble size effect.

Table 6.8 Performance of LSTM-RNN in ALT short tokens processing

model	Separate tokenization			Joint tokenization and POS-tagging				
	Accuracy (%)		F-score	Accuracy (%)		F-score		
	Test	(dev)	token	test	(dev)	token	(Δ_{joint})	token/POS
RNN ENS-5	94.8	(95.1)	0.942	93.7	(93.9)	0.945	(+0.003)	0.938
RNN ENS-10	94.9.	(95.1)	0.943	93.7	(93.9)	0.945	(+0.002)	0.938
RNN ENS-20	95.0.	(95.1)	0.943	93.8	(94.0)	0.946	(+0.003)	0.939
RNN ENS-50	95.0.	(95.1)	0.944	93.9	(94.1)	0.946	(+0.002)	0.939
RNN ENS-100	95.1	(95.1)	0.944	93.9	(94.1)	0.947	(+0.003)	0.940
RNN MAX @100	94.4	(94.6)	0.938	93.1	(93.3)	0.939	(+0.001)	0.931
RNN MIN @100	93.9	(94.2)	0.932	92.6	(92.9)	0.937	(+0.005)	0.928
CRF 2- IBES	94.9	_	0.943	93.9	_	0.947	(+0.004)	0.940
CRF 3- IBES	95.1	_	0.944	93.9	_	0.946	(+0.002)	0.940

Table 6.8 is the performance of LSTM-RNN of ALT short tokens processing with the accuracy scores of test and development data set. The ensemble (ENS-) results over 5, 10, 20, 50, and 100 models are also showed. The best (MAX@100) and the worst (MIN@100) single models among the total of 100 models are listed as well. The results of 2- and 3-IBES of CRF experiment in ALT short tokens are also listed in Table 6.8 for comparison.

Generally, LSTM-RNN boosted by ensemble can achieve performance comparable to that of CRFs on the full training data. The LSTM-RNN can provide better performance on small training data rather than the separate tokenization case for the case of joint tokenization and POS-tagging.

Therefore, specifically on small training data, it is noticeable that a more informative output tag set has more significant effects because the sparseness of the discrete features used in the CRFs is eased by the embedding to a compact representation in a low-dimension real space facility of RNN.

As for the effect of the ensemble, it can improve the performance of only a few models (e.g., five), although there is stable and gradual but insignificant improvement in the performance of a large ensemble number of models.

Table 6.9 presents the experimental results of ALT long tokens by LSTM-RNN.

Table 6.9 Performance of LSTM-RNN in processing long tokens of ALT

model	Separate tokenization			Joint tokenization and POS-tagging				
	Accuracy (%)		F-score	Accuracy (%)		F-score		
	test	(dev)	token	test	(dev)	token	(Δ_{joint})	token/POS
RNN ENS-5	95.6	(95.7)	0.933	94.1	(94.3)	0.934	(+0.001)	0.926
RNN ENS-10	95.6.	(95.9)	0.933	94.2	(94.5)	0.935	(+0.002)	0.928
RNN ENS-20	95.7.	(95.9)	0.934	94.3	(94.7)	0.935	(+0.001)	0.928
RNN ENS-50	95.7.	(96.0)	0.935	94.4	(94.7)	0.937	(+0.002)	0.930
RNN ENS-100	95.7	(96.0)	0.935	94.4	(94.7)	0.937	(+0.002)	0.930
RNN MAX @100	95.1	(95.4)	0.926	93.7	(93.8)	0.928	(+0.002)	0.920
RNN MAX @100	94.6	(95.0)	0.919	93.0	(93.4)	0.922	(+0.003)	0.913
CRF 2- IBES	95.5	_	0.932	94.3	_	0.937	(+0.005)	0.929
CRF 3- IBES	95.6	_	0.933	94.3	_	0.936	(+0.003)	0.929

.

From the above two groups of experiments, it can be perceived that LSTM-RNN can also perform better in joint tokenization and POS-tagging than in separate tokenization. The effect of RNN is more obvious than that of CRFs when processing on ALT long tokens. As the above discussions, low-order N-gram features cannot capture sufficient local information, while high-order N-gram features cause sparseness. Therefore, this dilemma can be assuaged by the strength of RNN, which can provide more efficient feature representation.

6.2 Experiments of Dependency Parsing for Myanmar Language

Dependency parsing is more useful to represent syntactic information for free word order languages than syntax parsing. As the lack of dependency resources and free word order nature of Myanmar language, dependency parsing by unsupervised approach was applied in building dependency treebank by using segmented and POS tagged corpus. Therefore, as a pilot experiment of unsupervised dependency parsing on raw segmented and part-of-speech (POS) tagged Myanmar sentences has been implemented and has been described in [p3]. Seven referenced word dependency schemes have been presented.

Being able to apply unsupervised dependency parsing for Myanmar sentences by UDPipe, all sentences from corpus were annotated by unsupervised way as the second step to get raw annotated dependency information for Myanmar sentences by nine dependency referenced schemes of phrases which are most occurred in Myanmar sentences and has been described in [p3].

6.2.1 Experimental Setting for Unsupervised Dependency Parsing with POS

UDPipe (UD 2.0) was set since it is a trainable pipeline and can perform segmentation, POS tagging, lemmatization and dependency parsing on CoNLL-U format data which uses universal POS (U-POS) tags. Shared Japanese model had been used as a training model because of having similar dependency structures as Myanmar grammar and no annotated resource for Myanmar dependency structure to use training data. The CoNLL-U Viewer had been used to review and check the results. Input segmented sentences were tagged with language general and U-POS tags in order to convert CoNLL-U format. Therefore, mapping scheme between Myanmar POS and U-POS tags has been described. The result unsupervised dependency parsed trees could be evaluated by unlabeled and labeled attachment scores (UAS and LAS) by UDPipe.

Statistics information of train and test data set used for evaluation analysis is presented in Table 6.10.

Table 6.10 Statistics on data set and evaluation scores

Data set	Sentences	Word Tokens	Evaluation scores	
			UAS (%)	LAS (%)
Train	5,113	110,985	98.25	97.89
Test	100	2,410	89.79	85.56

6.2.1.1 Results and Discussion of Unsupervised Dependency Parsing with POS

UAS and LAS accuracy scores of test and trained data had received 89.79 % and 85.56% and 98.25% and 97.89% respectively. Sample output tree and reference tree is shown in Figure 6.5 and 6.6.

The result tree of UDPipe needs to be annotated again as the referenced trees in order to provide more reliable syntactic and semantic information although the main root verb order of the result trees and most words structures like suffixed word tagged as PPM or ADP and PART are correct and acceptable.

However some word dependency structures are not correct due to different structure between Japanese and Myanmar words.

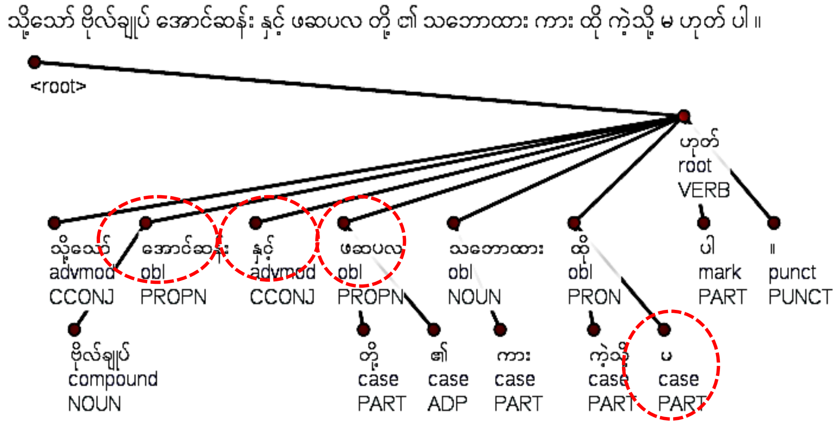


Figure 6.5 Output unsupervised dependency parsed tree of UDPipe

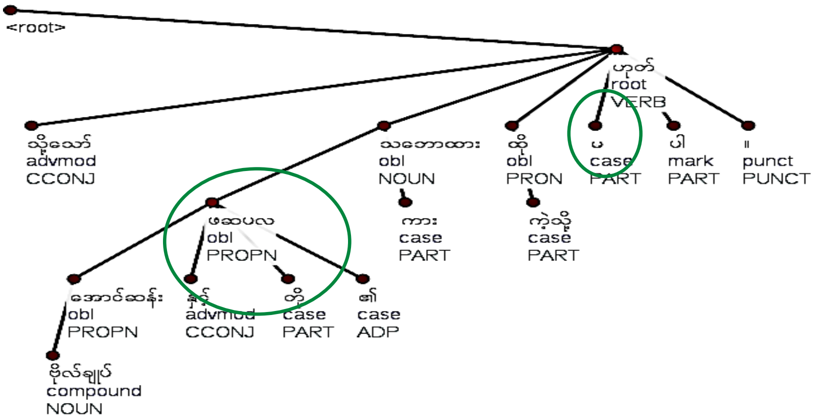


Figure 6.6 Referenced tree with dependency scheme

But unsupervised dependency parsing by UDPipe by applying language and U-POS tags information could be successfully applied and provide valuable outcome of this experiment could for faster annotation for dependency information of Myanmar sentences rather than manual ways. Moreover, it can be seen that accuracy scores of unlabeled and labeled attachment by UDPipe were relatively high.

6.2.2 Unsupervised Corpus Annotation

Unsupervised corpus annotation was done by two steps. As a first step, all sentences from the corpus had been parsed by shared Japanese model as the procedures described in above section due to the acceptable outputs generated from the pilot experiment by high accuracies of evaluation analysis. Then, unsupervised annotated dependency parsed trees by Japanese model were divided into train and test data.

Some training data were updated again according to the reference structures to evaluate unsupervised annotated sentences by a Myanmar model. Therefore, a new model was built by training data including some re-annotated data and data without re-annotating. Then the first model of Myanmar language was also evaluated by test sentences. The accuracies are measured by UAS and LAS [p3].

6.2.2.1 Results and Discussion of Unsupervised Corpus Annotation

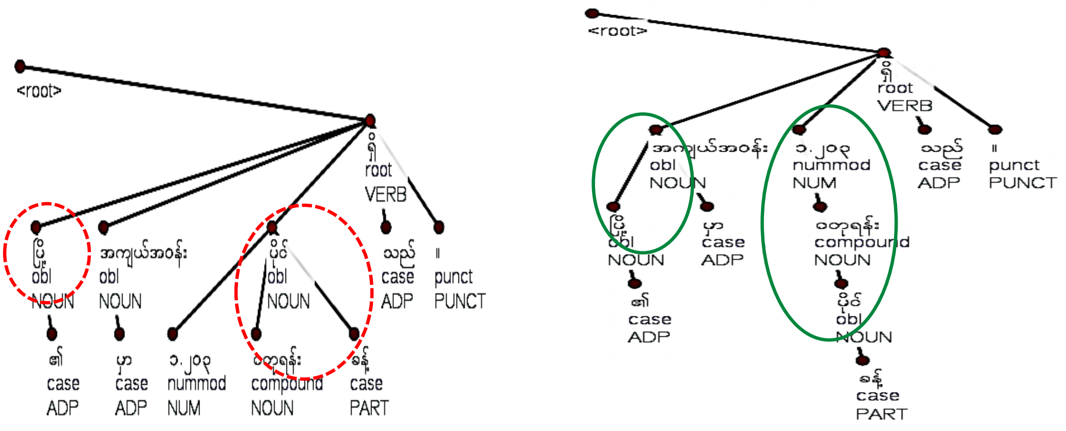
Table 6.11 shows the statistics of train and test data of the corpus and accuracies results by the model built after unsupervised corpus annotation [p3].

Table 6.11 Statistical information and results of annotated corpus

Data set	Sentences	Word Tokens	Accuracy	
			UAS (%)	LAS (%)
Train	10,000	217,636	93.88	92.57
Test	287	6,504	93.20	91.21

Based on the manual checking results by the reference schemes, PART, PPM, and V have high correct rates among other tags because of the similar suffixing styles and the grammar order between Myanmar and Japanese. However, the correct rates of CONJ, N, and ADJ were relatively low since Myanmar words and phrases are frequently combined to form a syntactic role and this effect causes wrong heads in dependency trees although Japanese model could provide correct heads of simple words or phrases or clauses

Anyway, generating correct root heads for sentences and frequently occurred suffixed words by both Japanese model and Myanmar model built by modified training data on unsupervised annotated results is very worthy to get dependency information for Myanmar Language since adding manually dependency structures to the segmented and POS tagged Myanmar sentences is complicated and very time-consuming task. Unsupervised annotated result tree is shown in Figure 6.7 by comparing with the reference tree on it.



(A) Unsupervised annotated tree **(B) Reference tree**

Figure 6.7 Example of unsupervised annotated tree and reference tree

Only the word nodes in circles of unsupervised tree do not attach as nodes in circles of the reference tree. Therefore, to be reliable dependency annotations in corpus, the result trees of unsupervised corpus annotation were checked and updated again with the reference dependency structures in post processing as mentioned in Section 4.6 in the Chapter 4.

6.2.3 Experiments in Post Processing

Manual checking and updating as the reference dependency structures in all unsupervised annotated sentences in the whole corpus in time is a difficult task. Therefore, in post processing, the selected unsupervised annotated sentences were checked and updated by reference dependency schemes firstly. As a second step of post processing, dependency parsing was carried out by the model with the combined data including updated data and not updated data. Then the post processing was carried out repeatedly by manual updating and dependency parsing method as a bootstrapping manner to reduce manual updating errors and be consistent updating as described in Section 4.6 in Chapter 4.

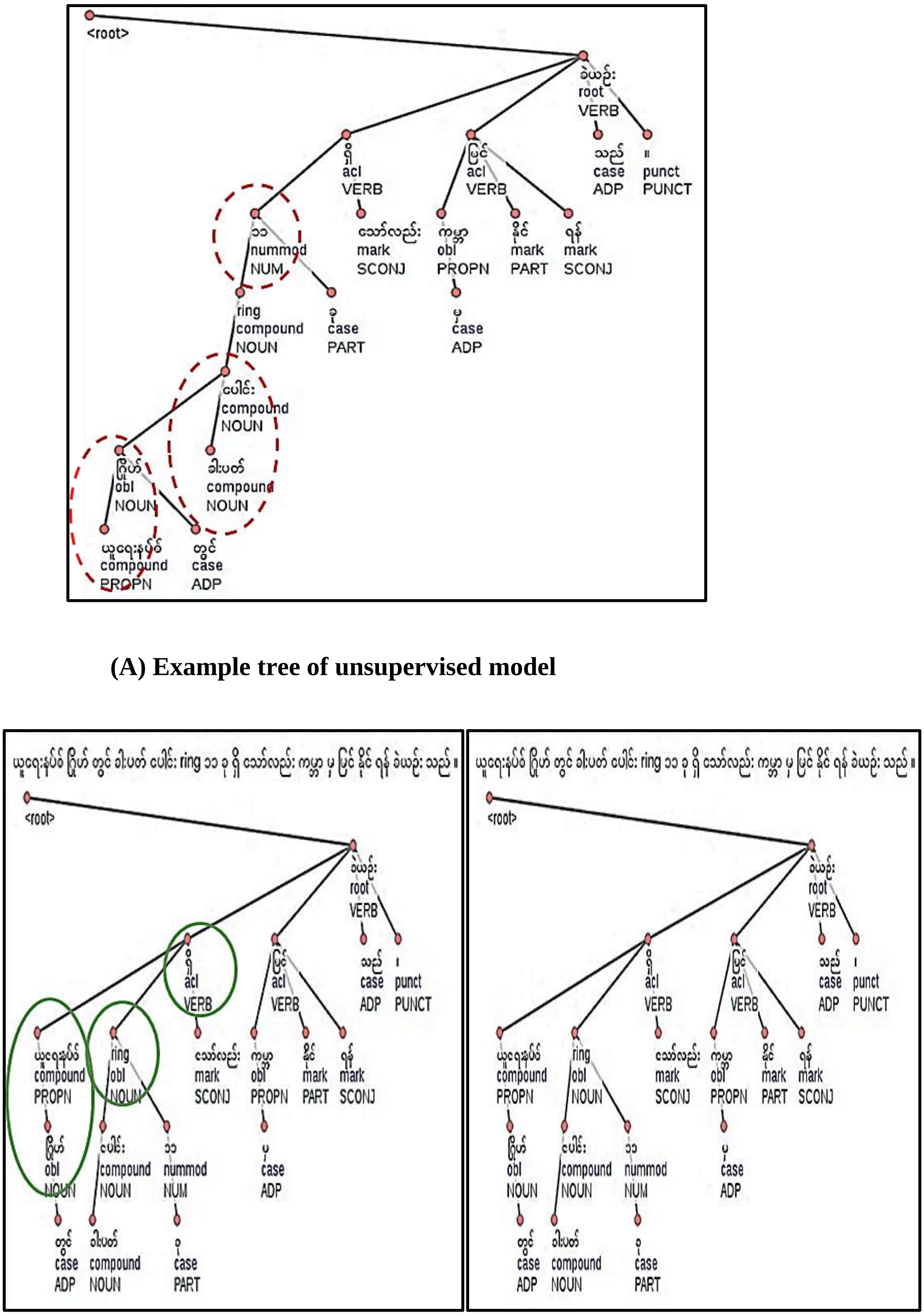
In order to investigate how to achieve similar annotation structures to reference dependency schemes by post processing, two dependency parsing experiments were implemented with the models built by UDPipe. The first experiment was implemented by results of unsupervised dependency corpus annotation before post processing. And the second experiment was executed by the post processed data.

6.2.3.1 Results and Discussion of Experiments in Post Processing

The result trees of the unsupervised annotated model will be compared with the result trees of the post processed model in order to know the influence of post processing on treebank data. Result trees of the models before post processing and after post processing are presented by short and long sentences.

Figure 6.8(A) is the example result of short sentence generated by the model before post processing. Figure 6.8(B) is the sample result of short sentence generated by the model after post processing. In short sentences, the word nodes in dotted circles such as proper noun phrase (“ယူရေးနပ်စ် ဂြိုဟ်”), compound noun phrase (“ခါးပတ် ပေါင်း *ring*”) do not attach related correct head node.

The example result of long sentence generated by the model by post processing model is shown in Figure 6.9 (A). In that tree, the word nodes, phrases and clauses of long sentences in dotted circles do not attach correct head nodes. And the example



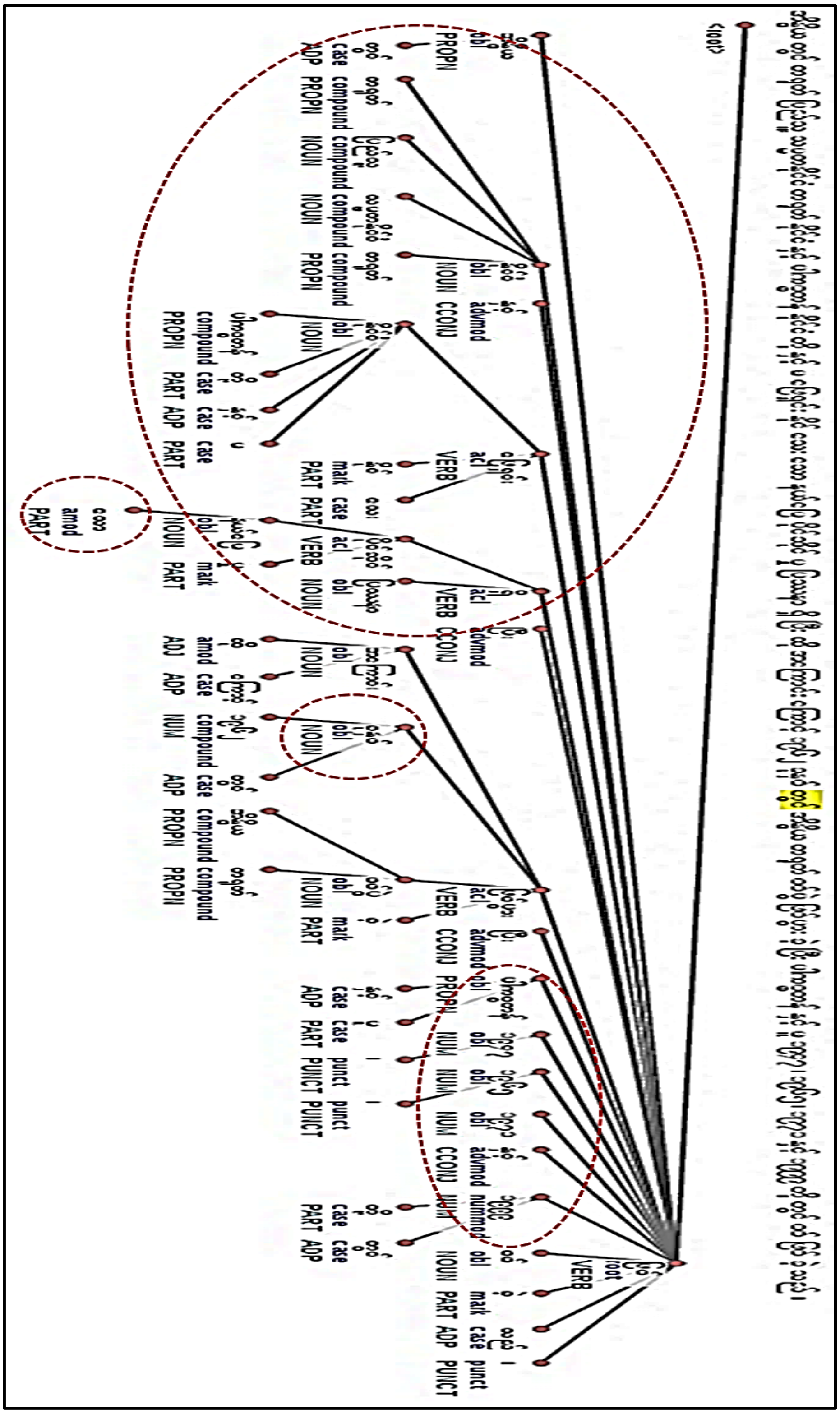
(A) Example tree of unsupervised model

(B) Example tree of post processed model

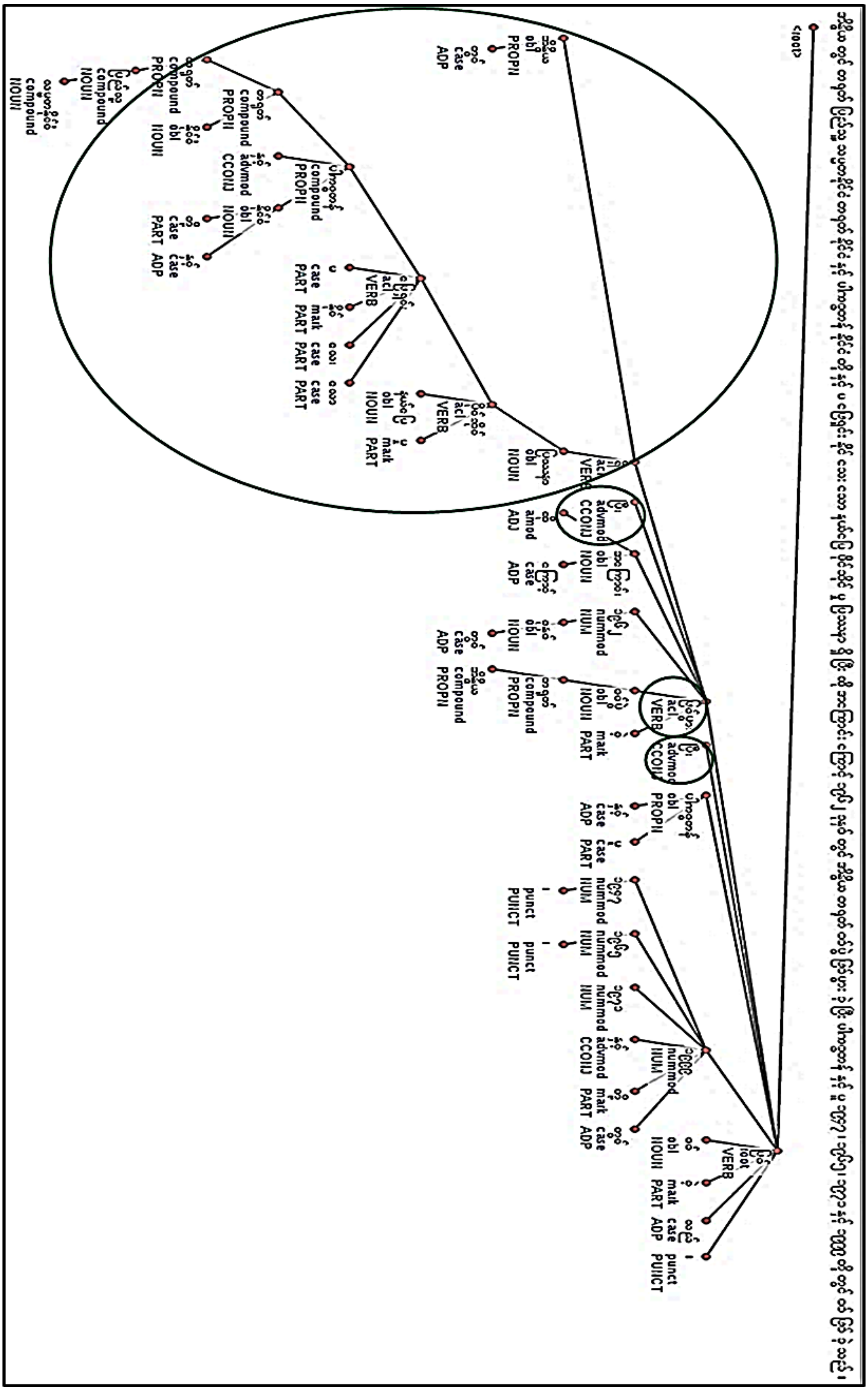
(C) Reference tree

Figure 6.8 Example trees of unsupervised and post processed models with reference tree for formal short sentence

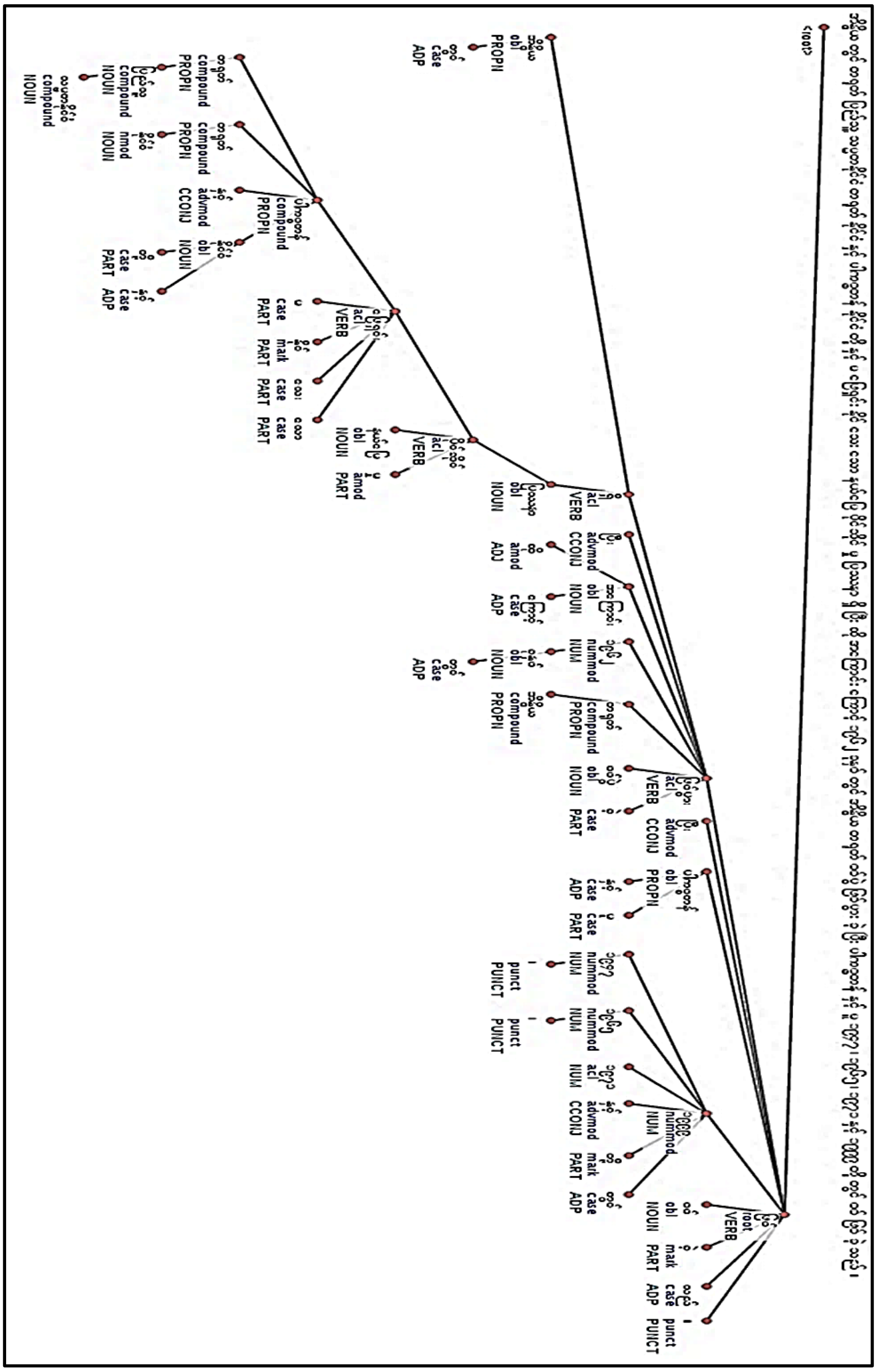
result of long sentence generated by the model after post processing is shown in Figure 6.9 (B). In that tree, the word nodes, phrases, and clauses attaching incorrectly in the unsupervised tree attached to more correct and related head nodes. The reference tree of example long sentence is shown in Figure 6.9(C).



(A) Example tree of long sentence by unsupervised model



(B) Example tree of long sentence by post processed model



(C) Reference tree

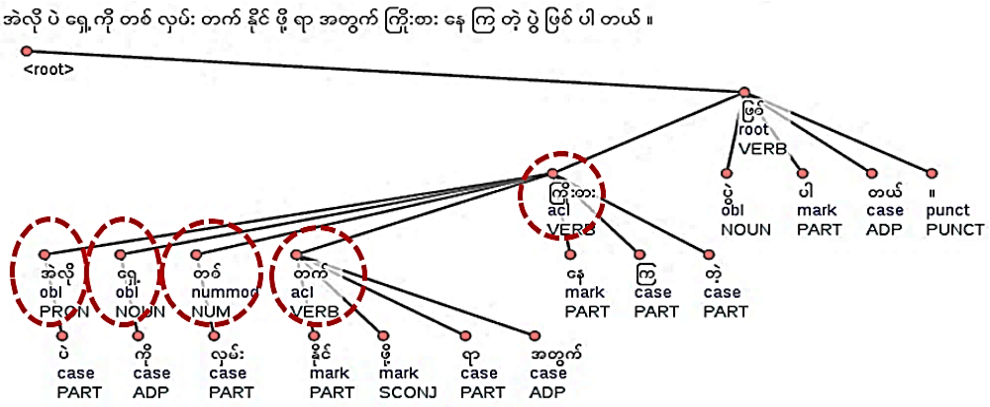
Figure 6.9 Example trees of unsupervised and post processed models with reference tree for long sentence

6.2.4 Experiment on Different Domain Sentences by Post Processing Model

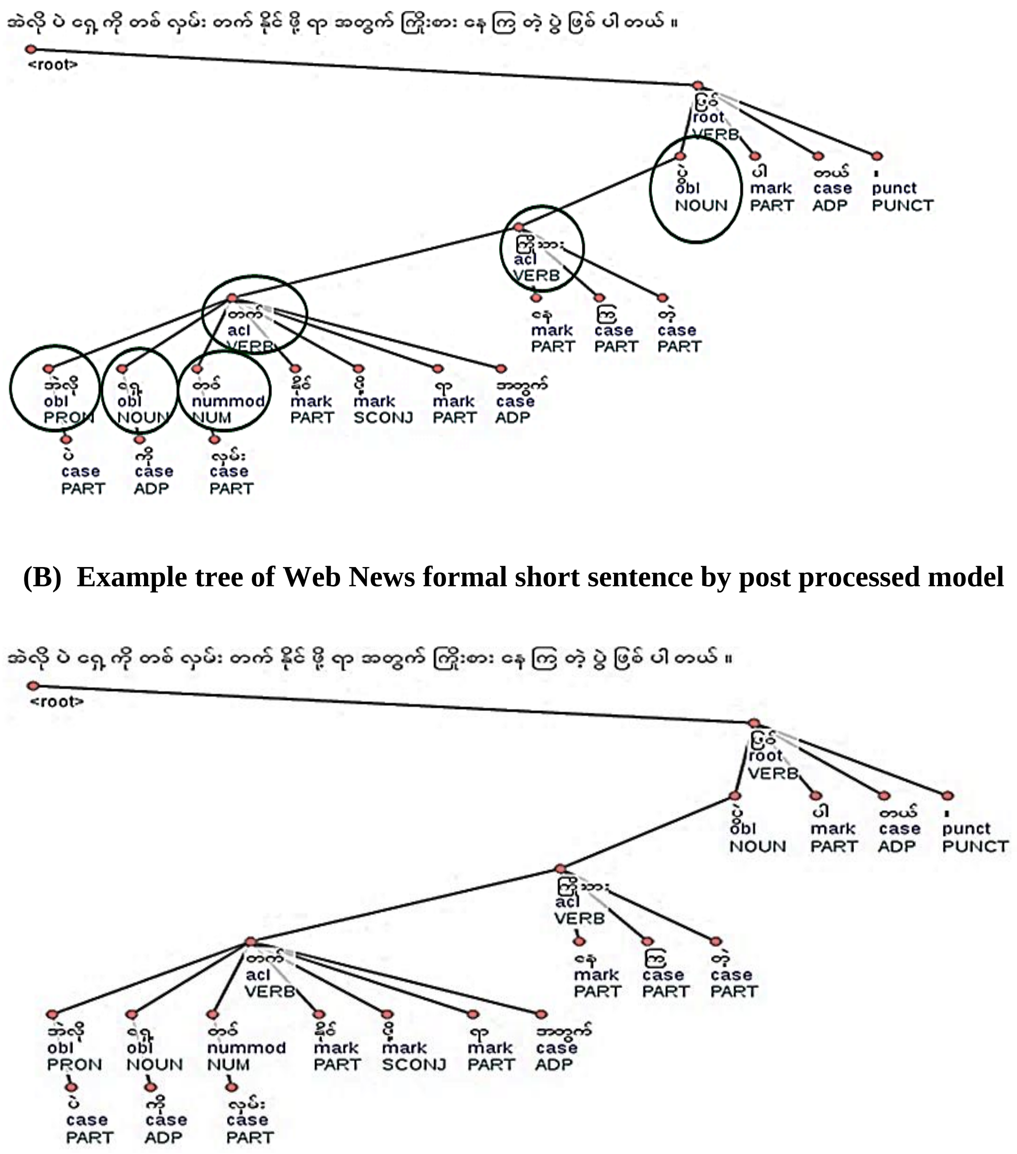
As the model with post processed data could produce close dependency tree like the reference dependency schemes in the above example cases, sentences from Web News domain corpus were parsed by the model with post processed sentences from general domain to add more sentences in Myanmar treebank without being time-consuming. More dependency scheme types in treebank can provide more reliable and acceptable parse tree and current modern Myanmar sentences can be written by free styles according to the Myanmar grammar structures. Therefore, this experiment was implemented to increase Myanmar treebank resource with more dependency structures of Myanmar grammars. The example result trees of the model built by unsupervised annotated data and the model built by post processed data will be compared in the following section.

6.2.4.1 Results and Discussion of Experiments on Different Domain Sentences

The first post processed data were general domain as mentioned in Chapter 4, the model built by updated post processed sentences could generate very similar dependency parsed tree as the reference dependency structures for formal short sentences .The tree generated by the model built by unsupervised annotated data for formal short sentence from different domain is shown in Figure 6.10 (A). In that tree, word nodes in circles do not attached to their related head nodes.



(A) Example tree of Web News formal short sentence by unsupervised model



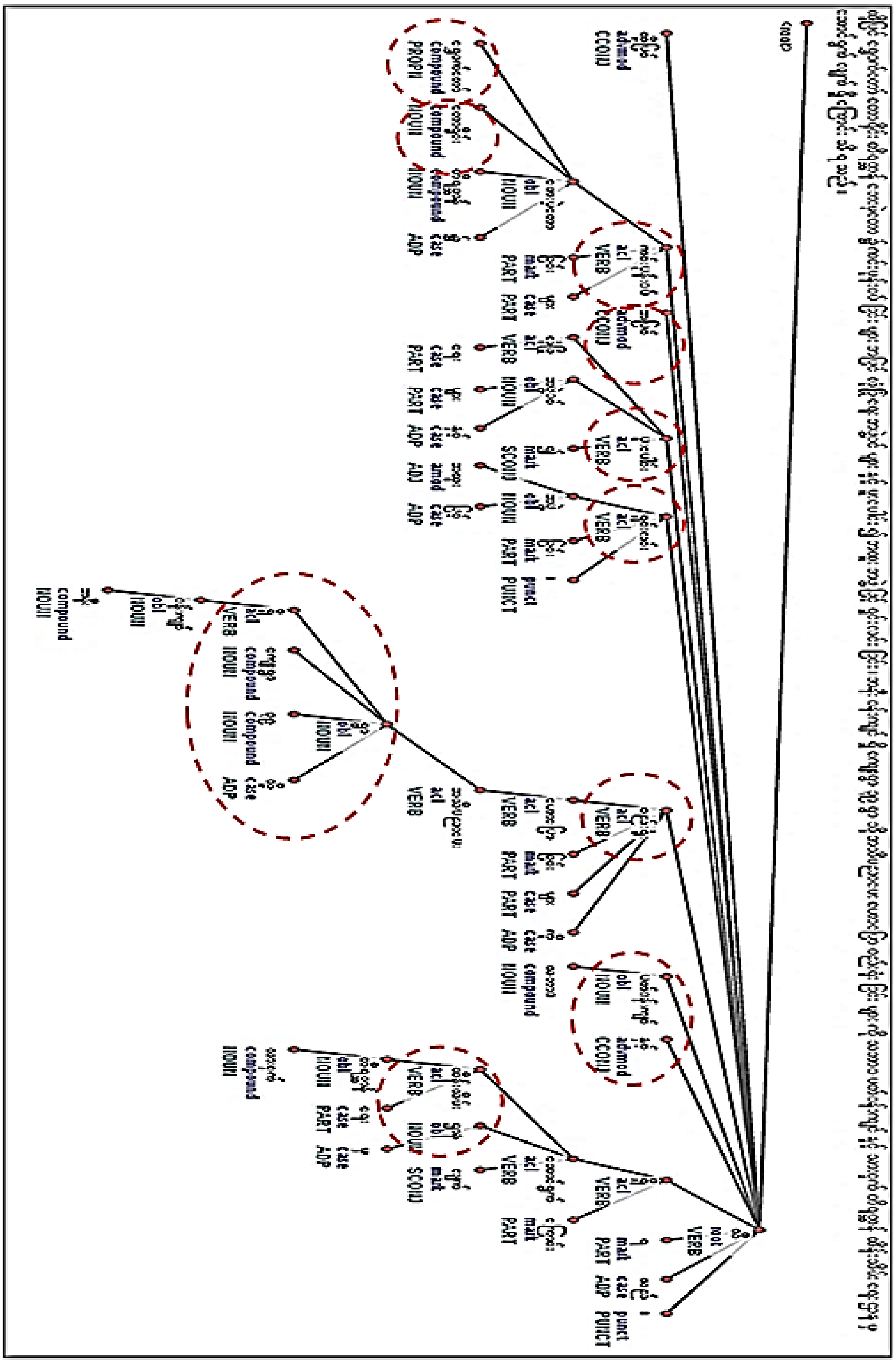
(B) Example tree of Web News formal short sentence by post processed model

(C) Reference tree of Web News formal short sentence

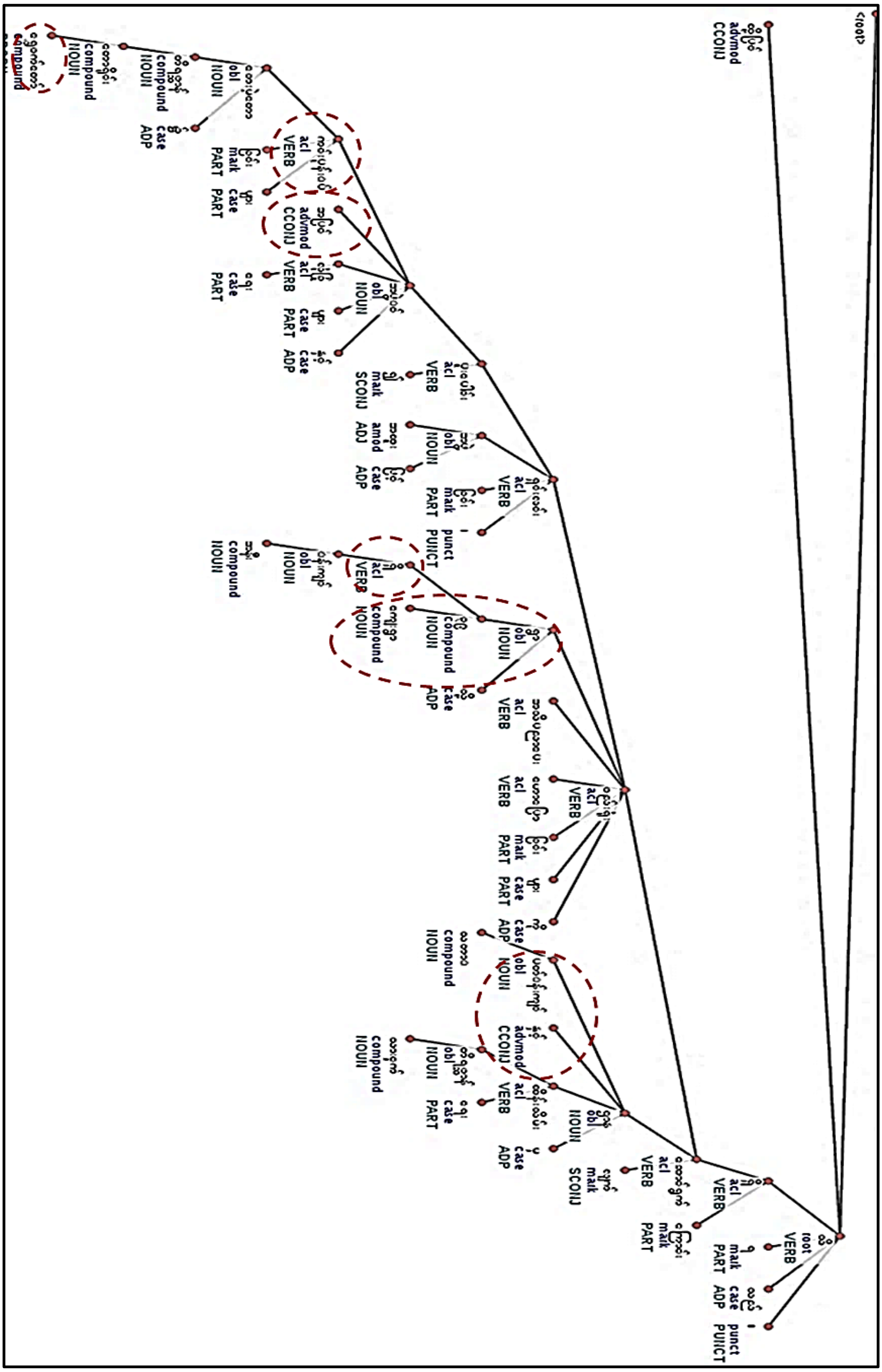
Figure 6.10 Example trees of unsupervised and post processed model with reference tree for Web News formal short sentence

However, in the Figure 6.10 (B), word nodes in circles of the result tree provided by the post processed model attach properly to their related nodes as the referenced tree shown in Figure 6.10 (C).According to the result trees of the above figures, the post processed model can generate similar parse tree for formal short sentences as the reference dependency scheme.

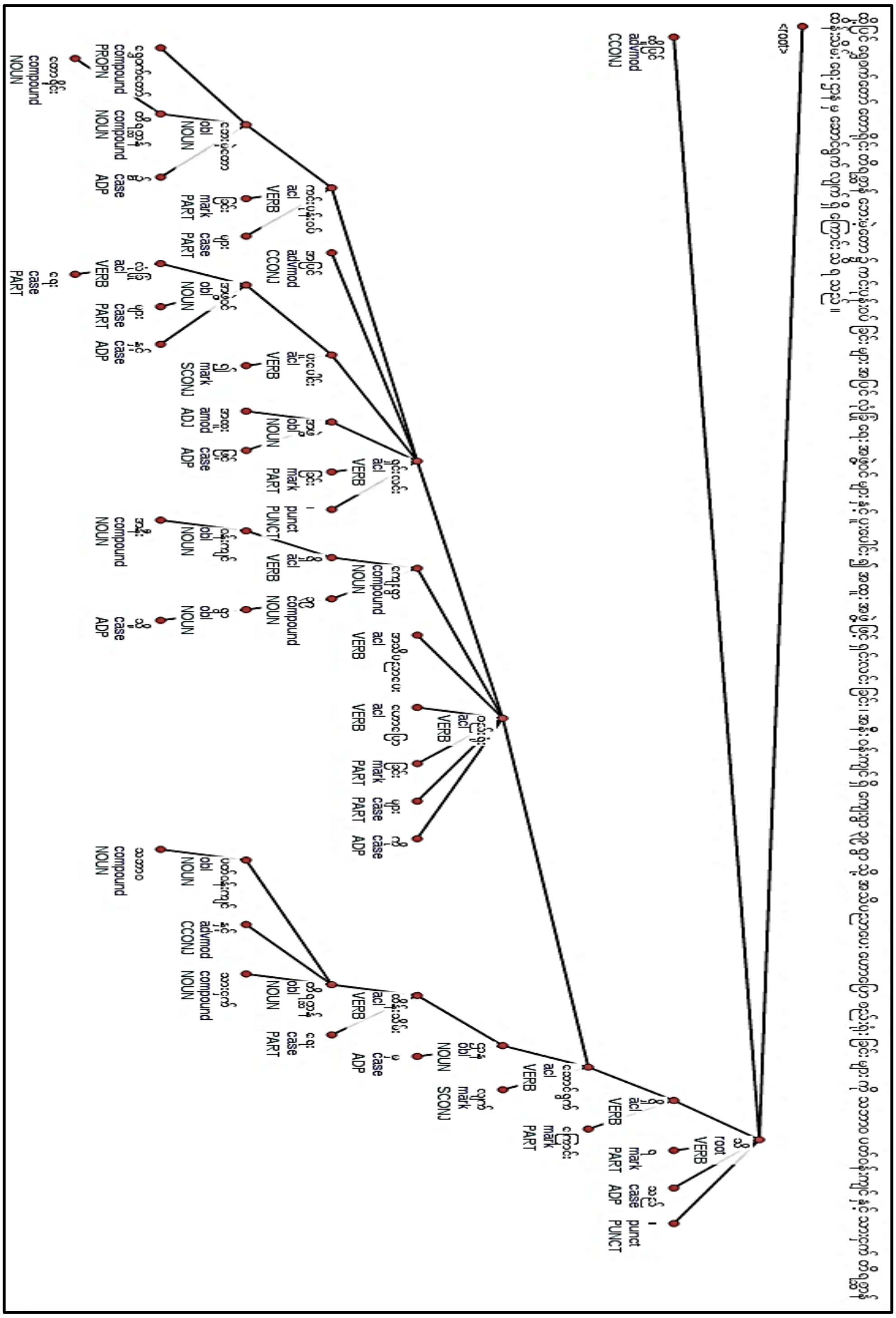
Moreover, it could generate better parse tree for long sentence from different domain shown in Figure 6.11. In Figure 6.11 (A), few nodes in circles do not attach to their related head nodes.



(A) Example tree of Web News long sentence by unsupervised model



(B) Example tree of Web News long sentence by post processed model



(C) Reference tree of Web News long sentence

Figure 6.11 Example trees of unsupervised model and post processed model with reference tree for Web News long sentence

It can be seen that the post processed model could generate better parse tree rather than result trees with more wrong circled nodes of the unsupervised annotated model shown in the above Figure 6.11 (A) although its parse tree is not as the reference tree shown in Figure 6.11 (C). Moreover, manual updating in result trees of post processed model to be correct dependency tree is easier and faster than updating in result of unsupervised annotated model.

According to the result trees generated post processed model for short and long sentences from same or different domain corpus, post processing on sentences of unsupervised dependency corpus annotation can well support to build dependency treebank construction with not only easy and fast updating time but also consistence updated data for Myanmar language.

6.3 Experiment settings for Evaluations of Myanmar Dependency Treebank

As the parsing performance evaluation of the developed dependency treebank, parsing experiments have been implemented to measure the performance of the parsing models built by treebank data and has been described in [p5]. As the treebank contains three different domains corpora, cross validation testing was used for the evaluation of the performance of each domain-wise parsing models for Myanmar dependency parsing since cross validation testing is mainly used in settings where the goal is prediction, and one wants to estimate how accurately a predictive model will perform in practice. Train and test data partition in k-fold cross validation can be illustrated as in Figure 6.12.

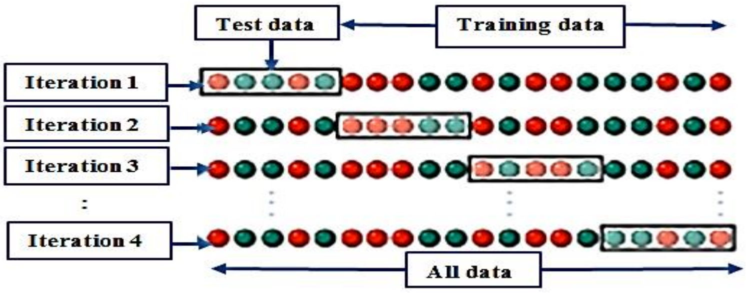


Figure 6.12 K-fold Cross Validation

For parsing experiments on different domain corpora of treebank, 90% and 10% of total sentences from each domain corpus of treebank were divided for training and test data set to evaluate by 10-fold cross validation test. Alternate range order of all data in each corpus for test sentences and all the rest for train data in each corpus were split automatically into equally-sized test and train data parts for each validation test by python script.

Then parsing experiments have been done repeatedly by 10-fold partition of train and test data for each domain corpus.

After finishing cross validation tests on all corpora, the average score of all validation tests have been calculated to report as total average score of cross validation test for each domain corpus.

For the parsing performance of the whole total data of Myanmar dependency treebank, parsing experiments have been done by the model building with all post processed data of treebank.

All parsing experiments were implemented by UDPipe 1.2 which is an open-source trainable pipeline processing tool which can performs sentence segmentation, tokenization, POS tagging, lemmatization and dependency parsing without the need for any other external data for multiple languages. And it is also available under open-source Mozilla Public License (MPL) and provides bindings for C++, Python, Perl, Java and C#. Evaluation scores are measured by UAS and LAS scores.

6.3.1 Experimental Results of Cross Validation

The results of parsing performance experiments on each corpus of treebank data are evaluated by 10-fold cross validation tests as described in above. In each 10-fold cross validation tests, 10,010 sentences, 9,000 sentences, and 1,620 sentences were used for the training data of myPOS, ALT, and Web News corpus respectively. And 1,000 sentences, 1,000 sentences, 180 sentences from the alternate test range of myPOS, ALT, and Web News corpus were used respectively as test data in each 10-fold cross validation test. For the cross validation tests of each corpus, the average sores are measured by unlabeled attachment score (UAS) and label attachment score (LAS) as parsing experiment scores and shown by each corpus in Table 6.12.

Table 6.12 Average parsing scores on domain-wise cross validation tests

Corpus	**Average Scores**	
	UAS (%)	**LAS (%)**
myPOS	89.12	86.84
ALT	91.40	90.24
Web News	87.03	84.83

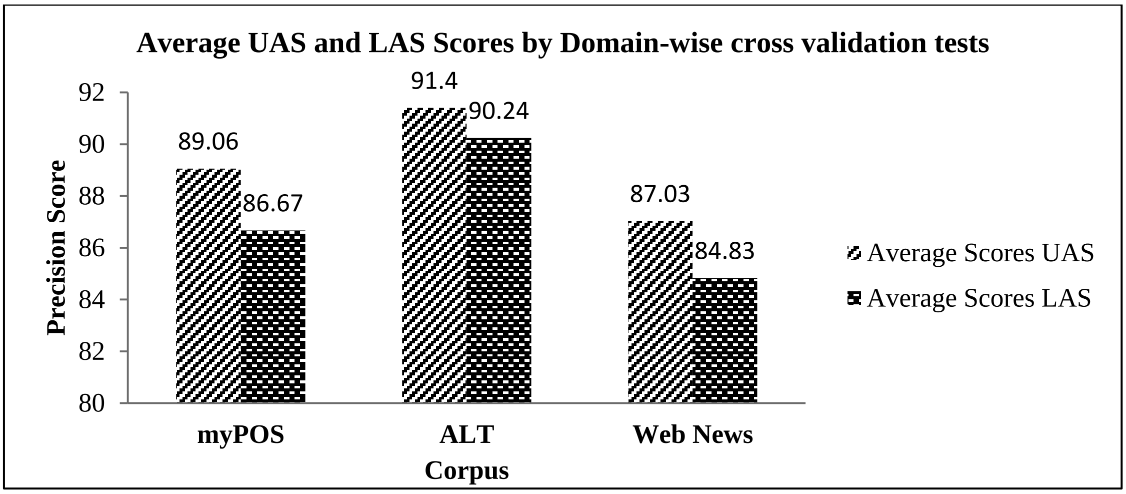


Figure 6.13 Average parsing scores on domain-wise cross validation tests

Figure 6.13 shows the graph of average UAS and LAS scores as the parsing performance evaluation results of each corpus in treebank by 10-fold cross validation tests.

6.3.2 Experimental Results on each Corpus of Treebank

In order to investigate accuracy score based on the size and types of training data and the performance of the model built by total data of Myanmar treebank, parsing experiments by selected test sentences from each different corpus of treebank were executed by the trained model created by combining 22,810 sentences form all domain data of treebank. Selected data of each corpus were used as closed test data in experiments and the accuracies scores of each corpus in these experiments by the multi-domain model are shown in Table 6.13. They are also shown in Figure 6.14 by graph to be able to check easily and quickly.

Table 6.13 Experimental results by multi-domain model

Test Corpus	Test Sentence	UAS (%)	LAS (%)
myPOS	1,000	91.36	90.08
ALT	1,000	92.80	91.83
Web News	180	87.97	86.25

In Figure 6.14, all result scores of selected test sentences from each corpus are slightly higher than average scores of domain-wise cross validation tests because the training model size and sentence types are larger and more than domain-wise models. The more training data size provides the higher accuracies score. However, it is needed to be unique annotation format in treebank for all sentences although their domains are different. To be the unique annotation format for different domain data is significant time-consuming task.

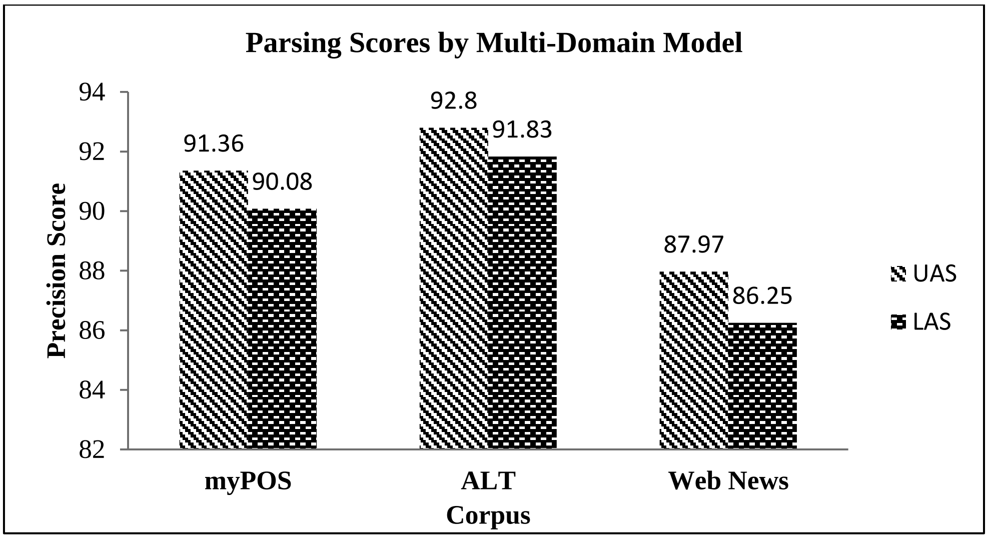


Figure 6.14 Parsing results by multi-domain model

6.3.3 Evaluation Results of Treebank

The overall evaluation on parsing treebank data were also analyzed by evaluating on each corpus data by the “conll17_ud_eval.py”, CoNLL 2017 UD parsing evaluation script, in order to know detailed parsing accuracy scores of each corpus. Gold standard file and system output file are needed to input to evaluation script. For the overall evaluation of parsing treebank data, post processed data from two corpora: myPOS and Web News and unsupervised annotated ALT data of treebank are used as gold standard files. To generate system output of each corpus, the multi-domain model was used as a system model since it has been built by combining all data of three different domain corpora of treebank.

Then, three post processed corpora of treebank have been parsed by the multi-domain model to get system parsing outputs. After generating system parsed outputs of three corpora, these system parsed results have been compared with dependency head nodes of three corpora of treebank assuming as gold standard annotated files for evaluation accuracy status of all system generated parsed trees. The parsing evaluation accuracy scores for the occurrence of head nodes in matching system output with data of all corpora of current Myanmar dependency treebank are presented in Figure 6.15.

As the evaluated parsing precision scores for the whole data from three corpora of current Myanmar dependency treebank, the two post processed corpora: myPOS and Web News, have achieved over 87%, and 85% for UAS and LAS respectively in matching up system generated data of them with manual post processed reference data and the automatic annotated ALT corpus has achieved over 92% and 91% for UAS and LAS respectively in matching up system generated data of it with automatic unsupervised annotated reference data.

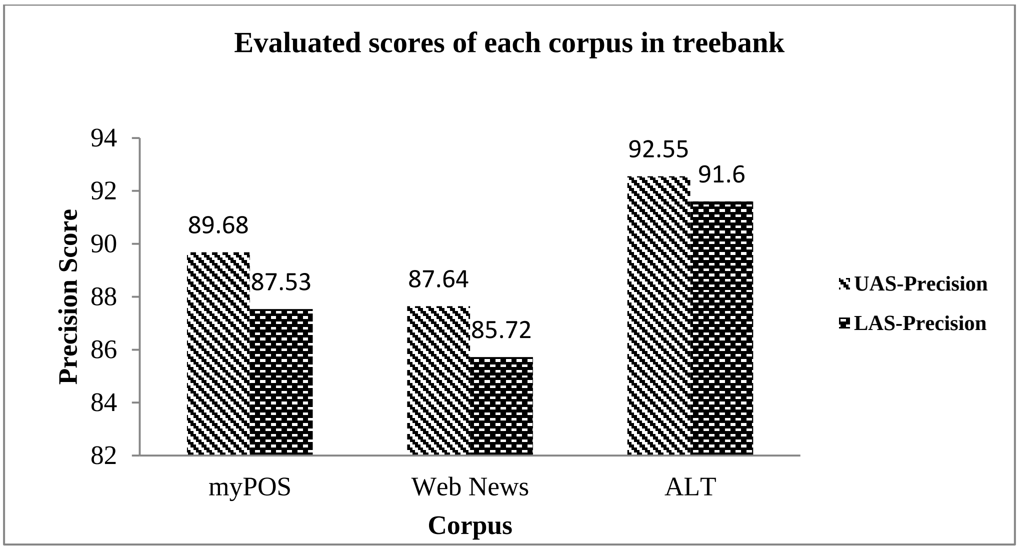


Figure 6.15 Evaluation results on corpora of treebank

Evaluation scores of unsupervised annotated ALT corpus is more than manual post processed data of myPOS and Web News corpus since sentences of ALT are longer than sentences of myPOS and Web News. On the other hand, as system generated outputs, outputs of myPOS and Web News corpus are more similar to reference dependency structures than outputs of ALT in manual random checking on system generated trees [p5].

6.4 Chapter Summary

In this chapter, all results of experiments done during the whole research have been described. And also the detailed discussions on the evaluation results of experiments have also been presented. Moreover, the experimental results of Myanmar parsing model built by updated post processed data have been presented. Indeed, the detailed discussions of evaluation results of the parsing model built by updated Myanmar dependency treebank data have been presented.

CHAPTER 7

CONCLUSION AND FUTURE WORK

Unsupervised dependency parsing by transition-based dependency parsing on transition predictions of neural network classifier model for Myanmar language has been proposed since parsing Myanmar sentence is also still a challenging task and Myanmar language is also free word order and agglutinative language. This proposed work is the first work for Myanmar language as far as we know. This chapter presents the summary of the research work with the privileges and limitation scope of it.

The main contribution of this research is proposing unsupervised dependency parsing by applying Universal Dependencies and Universal POS tag scheme via language POS. The next contribution is applying unsupervised dependency parsing to annotate Myanmar dependency treebank as a bootstrapping way to be fast manual annotation and reduce inconsistent errors occurred in manual annotation. Moreover, Myanmar dependency structures have been defined by unsupervised parsing approach as the next contribution. The next one is checking unsupervised annotated dependency structures as post processing step to be reliable dependency information for Myanmar dependency treebank. The next contribution is building the first Myanmar dependency parsing model which is used to parse new input sentence in Myanmar dependency parsing based on transition predictions of neural network classifier. This research work is the first for building dependency treebank resource and parsing model and applying Universal Dependencies for Myanmar language to the best of our knowledge.

7.1 Summary

Annotating dependency information of Myanmar sentences and building Myanmar dependency treebank in this research is the first for Myanmar Language. This research has been done by two main parts: training and testing. To implement the training part, segmented and POS tagged corpus is needed. In training part, firstly, corpus cleaning process has been carried out before data preprocessing for training data since different domain corpora which have been segmented and POS tagged according to specific purposes used in treebank construction. Therefore, it is needed to be a unique format in word segmentation and POS tagging in corpora for building treebank.

To be unique POS tagging scheme, a new POS tagging scheme for Myanmar language relating to Universal POS tags and a mapping scheme between the two tag sets have been defined in Chapter 4 as a contribution of this research. In order to apply unsupervised dependency parsing and universal dependencies, all corpora used in treebank are needed to convert CoNLL-U format with specific language POS and related U-POS tags before annotation.

To annotate Myanmar dependency treebank, semi-automatic annotation way has been proposed to reduce cost, time, and inconsistent annotation because expensive and time-consuming manual treebank annotation might have inconsistent annotations and automatic dependency parsing and manual updating for post processing have been applied in treebank building.

The transition based dependency parsing of data driven approach has been proposed for Myanmar dependency parsing because not only phrase order in most Myanmar sentences is free order style but also one syntactic role of Myanmar sentences can be composed of one or more words or phrases or clauses. Therefore, predicting word sequences of sentence is importance for Myanmar dependency parsing. In data driven approach, transition based dependency parsing induces a model for predicting the transition from the given the transition history of transition configuration process in order to choose optimal transition of next state for pars tree. Thus, transition-based parsing approach based on predictions of neural networks classifier is proposed for Myanmar dependency parsing.

UDPipe is a pipeline processing tool to perform tokenization, POS tagging, parsing with CoNLL-U format. UDPipe also uses transition-based dependency parsing approach. For transition predictions of UDpipe parsing method, the neural networks classifier model is used.

CoNLL-U format corpora are annotated by UDPipe by using Japanese shared model as presented in Chapter 4 since there is any annotated dependency resources for Myanmar language currently and Japanese grammar structure is similar to Myanmar grammar to add raw dependency information automatically for Myanmar treebank.

To be reliable dependency information, reference word dependency schemes were defined according to Myanmar grammar book published by Myanmar language committee as one contribution of this research. Dependency head words of Myanmar sentences resulting from the unsupervised annotation were updated manually in post processing by reference dependency schemes. Post processing is recursively done by unsupervised dependency parsing as a bootstrapping way in order to reduce inconsistent error in manual updating and to be fast post processing.

Then, the first Myanmar dependency treebank has been annotated with three different domain corpora.

For the parsing performance of same domain corpus in treebank data, ten-fold cross validation tests have been implemented. Ten-fold cross validation tests have been executed on three corpora of Myanmar treebank by building the training models by alternate data range of each corpus to evaluate parsing performance of each same domain corpus. Moreover, parsing performance of total data of Myanmar treebank has been evaluated with the experiments by the multi-domain model. Parsing precision scores of experiments are measured by UDPipe in terms of two dependency parsing measurement scores: unlabel and label attached scores, LAS and UAS. Finally the detailed discussions of all experiments done during the whole research have been also presented.

All experiments in cross validation tests provided average accuracy scores with over 87% and 84% for UAS and LAS scores for the same domain open test data by same domain-wise models. For the experiments with close test data of each corpus by the multi-domain model built by three difference domain corpora: myPOS, Web News, and ALT, of treebank, parsing experiments with multi-domain model achieved accuracy scores with over 87% and 86% for UAS and LAS for selected test sentences from each corpus respectively.

Post processed model generated better result trees than results of unsupervised model. Example results of two models have been presented and also discussed in Section 6.2.3 of Chapter 6. It is difficult to construct treebank containing all possible sentence structure types by supervised methods because of the issues of Myanmar sentences discussed in Chapter 3 and Chapter 4, the special free word order nature and fewer limitations in grammar rules of Myanmar language than those of other languages as English.

In conclusion, the first Myanmar dependency parsing model has been built by Myanmar dependency treebank data created in this research. It can also provide acceptable dependency parse trees with high accuracy scores by the proposed transition-based dependency parsing based on predictions of neural networks classifier. Moreover, this work can provide beneficial information for direct dependency parsing for Myanmar sentences by Myanmar model in future and it is the first work for Myanmar language and.

7.2 Advantages and Limitations

Unsupervised dependency parsing has been effectively applied for Myanmar language, being low resource language up to now. This proposed unsupervised dependency parsing has supported building first dependency treebank resource and dependency parsing model for Myanmar language since there are still no dependency resources for Myanmar sentences currently.

As a result of Myanmar dependency parsing model, Myanmar sentences can be parsed directly by Myanmar model.

The proposed dependency parsing model parsed well most sentences and produced correct or acceptable parsed tree. Training data includes sentences from Wikipedia articles including history, economics, news, philosophy and politics areas, short conversations, news and news articles of news web sites. Interactive input sentence can be formal or informal types since there have been various sentence writing styles in Myanmar language. Unsupervised dependency labels of sentences in current Myanmar dependency treebank have not been post processed yet.

However, the proposed Myanmar parsing model can parse well new input sentences and result parsed trees are better than results of pure unsupervised dependency parsing with shared Japanese model. It can produce automatically their related dependency structures. Performance evaluations of the proposed Myanmar parsing model and dependency treebank data have been described in Section 6.3 of Chapter 6.

As a limitation in this research, it takes longer time to annotate longer sentences with various writing styles than short sentences. Therefore, adding more dependency trees to training data is not fast if there are few annotators as it is also needed to check not only in word segmentation and POS tagging but also in post processing on unsupervised parsed results. Training data must be a balance between short and long sentences data to provide better results.

7.3 Future Works

Although current Myanmar parsing model can parse well new input Myanmar sentences, it is needed to increase more data from news domain to be more correct grammar syntactic structures and more syntactic dependency forms of current Myanmar sentence written styles to Myanmar treebank in order to be more correct and reliable dependency parsed trees.

To be full and exact dependency information of sentences, it is needed to update with the annotation of full dependency labels instead of unsupervised dependency labels in treebank.

Myanmar dependency tree parsing can be improved by using different machine learning approaches and annotating complete dependency information to Myanmar treebank that will enhance Myanmar NLP research works in future since parsing is one main important part of natural language processing.

Parsing experiments have also been carried out for the performance of treebank data and results have been also presented. As a conclude, contribution is that this dependency head annotation for dependency treebank is the first work for Myanmar language and can provide useful information for direct dependency parsing for Myanmar sentences by Myanmar model in future.

AUTHOR'S PUBLICATIONS

[p1] H. T. Z. Aye, C. Ding, Win Pa Pa, K. T. Nwet , M.Utiyama , E. Sumita, "English-to-Myanmar Statistical Machine Translation Using a Language Model on Part-of-Speech in Decoding", 15th International Conference on Computer Applications (ICCA2017), pp.409-415, February 2017.

[p2] H. T. Z. Aye, Win Pa Pa, Y. K. Thu, "Unsupervised Dependency Parsing for Myanmar Language using Part-of-Speech Information" , 16th International Conference on Computer Applications (ICCA2017), pp.209-216, February 2018.

[p3] H. T. Z. Aye, Win Pa Pa, Y. K. Thu, "Unsupervised Dependency Corpus Annotation for Myanmar Language", 21st Conference of Oriental COCOSDA-International Conference on Speech Database and Assessments (2018), pp. 78 -83. IEEE, May 7-8, 2018.

[p4] C Ding , H. T. Z. Aye, Win Pa Pa, K. T. Nwet , K. M. Soe, M.Utiyama , E. Sumita, "Towards Burmese (Myanmar) Morphological Analysis: Syllable-based Tokenization and Part-of-speech Tagging", ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP) Journal, Volume 19 Issue 1, Article No. 5, June 2019.

[p5] H. T. Z. Aye, Win Pa Pa, "Dependency Head Annotation for Myanmar Dependency Treebank", Special Issue on Multidisciplinary Sciences and Engineering in Advances in Science, Technology and Engineering Systems Journal, Volume 5 Issue 6, pp. 788-800, November 2020.

BIBLIOGRAPHY

[1] A. Stolcke, "SRILM–an extensible language modeling toolkit," In the Proceedings of Seventh International Conference on Spoken Language Processing (ICSLP) 2002, pp. 901-904, 2002.

[2] C. Bosco, M. Simonetta, S. Maria, "Converting Italian Treebanks: Towards an Italian Stanford Dependency Treebank," In the 7th Linguistic Annotation Workshop and Interoperability with Discourse, pp.61- 69. The Association for Computational Linguistics, 2003.

[3] C. Ding, M. Utiyama, and E. Sumita, "NOVA: A feasible and flexible annotation system for joint tokenization and part-of-speech tagging," In the Journal of ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP), Volume.18, No.2 , pp. 17, December 2018.

[4] C. Ding, Y. K. Thu, M. Utiyama, A. Finch, E. Sumita, "Empirical dependency-based head finalization for statistical Chinese-, English-, and French-to-Myanmar (Burmese) machine translation," In the Proceedings of Workshop on Spoken Language Translation (IWSLT) 2014, pp. 184-191, 2014.

[5] C. Ding, Y. K. Thu, M. Utiyama, and E. Sumita, "Parsing Myanmar (Burmese) by using Japanese as a pivot," In the Proceedings of 14th International Conference on Computer Applications , pp. 158-162, February 2016.

[6] C. Gómez-Rodríguez, and J. Nivre, "Divisible transition systems and multiplanar dependency parsing," In the Journal of Computational Linguistics, IEEE, Volume. 39, No. 4, pp. 799-845, December 2013.

[7] C. Mărănduc, and CA. PEREZ, "A Romanian dependency treebank," In the International Journal of Computational Linguistics and Applications, Network and System Sciences, Volume. 6, No.2, pp. 83-103, 2015.

[8] C. Tillmann, "A unigram orientation model for statistical machine

translation," In the Proceedings of HLT-NAACL 2004: Short Papers, pp. 101-104. Association for Computational Linguistics, May 2004.

[9] D. Bamman, and G. Crane, "The Ancient Greek and Latin Dependency Treebanks," In the Proceedings of Language technology for cultural heritage 2011, pp. 79-98, 2011.

[10] D. Chen, and C. Manning, "A fast and accurate dependency parser using neural networks," In the Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp. 740-750, October 2014.

[11] D. Das, and S. Petrov, "Unsupervised part-of-speech tagging with bilingual graph-based projections," In the Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1, pp. 600-609. Association for Computational Linguistics, June 2011.

[12] D. Kingma, and J. Ba, "Adam: A method for stochastic optimization," In the Proceedings of the ICLR, December 2014.

[13] D. Mareˇcek, "Twelve Years of Unsupervised Dependency Parsing," In the Proceedings of ITAT 2016, CEUR Workshop Proceedings Vol. 1649, pp. 56–62, 2016.

[14] D. McClosky, E. Charniak, and M. Johnson, "Reranking and self-training for parser adaptation," In the Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics, pp. 337-344. Association for Computational Linguistics, July 2006.

[15] D. Zeman, "Reusable Tagset Conversion Using Tagset Drivers," In LREC 2008, Volume. 2008, pp. 28-30, May 2008.

[16] Department of the Myanmar Language Commission, Ministry of Education, "Myanmar Grammar (2013 Edition)," Published by Department of the Myanmar Language Commission, Ministry of Education, The Republic of the Union of Myanmar, 2013.

[17] F. J. Och and H. Ney, "A systematic comparison of various statistical alignment models," In the Journal of Computational

Linguistics, Volume. 29, No. 1, pp. 19-51, March 2003.

[18] F. J. Och, “Minimum error rate training in statistical machine translation,” In the Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1 Published by Association for Computational Linguistics , pp. 160-167, July 2003.

[19] G. Kikui, E. Sumita, T. Takezawa, and S. Yamamoto, “Creating corpora for speech-to-speech translation,” In the Proceedings of Eighth European Conference on Speech Communication and Technology (EUROSPEECH) 2003, pp. 381–384, 2003.

[20] G. Neubig, C. Dyer, Y. Goldberg, A. Matthews, W. Ammar, A. Anastasopoulos, M. Ballesteros, D. Chiang, D. Clothiaux, T. Cohn, K. Duh, M. Faruqui, C. Gan, D. Garrette, Y. Ji. Kong, A. Kuncoro, G. Kumar, C. Malaviya, P. Michel, Y. Oda, M. Richardson, N. Saphra, S. Swayamdipta, and P. Yin, “DyNet: The dynamic neural network toolkit,” In the Journal of arXiv preprint arXiv:1701.03980, January 2017.

[21] H. Isozaki, T. Hirao, K. Duh, K. Sudoh, and H. Tsukada, “Automatic evaluation of translation quality for distant language pairs,” In the Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing Published by Association for Computational Linguistics, pp. 944-952, October 2010.

[72] H. Yamada and Y Matsumoto, “Statistical dependency analysis with support vector machines,” In Proceedings of the 8th International nternational Conference on Parsing Technologies, pp. 195–206, April 2003.

[22] J. Blitzer, R. McDonald, and F. Pereira, “Domain adaptation with structural correspondence learning,” In the Proceedings of the 2006 conference on empirical methods in natural language processing (EMNLP), pp. 120-128. Association for Computational Linguistics, July 2006.

[23] J. Chun, NR. Han, JD. Hwang, and JD. Choi, “Building Universal Dependency Treebanks in Korean,” In the Proceedings of the Eleventh International Conference on Language Resources and

Evaluation (LREC 2018), May 2018.

[24] J. Nivre, “Dependency grammar and dependency parsing,” In the Journal of MSI report, Volume. 5133, No. 1959, pp. 1-32, 2005.

[25] J. Nivre, “Graph-Based and Transition-Based Dependency Parsing,” [Online].
Available:
http://ufal.mff.cuni.cz/~bejcek/parseme/prague/Nivre2.pdf

[26] J. Nivre, “Incrementality in deterministic dependency parsing”, In Proceedings of the Workshop on Incremental Parsing: Bringing Engineering and Cognition Together (ACL), pp. 50-57. Association for Computational Linguistics, July 2004.

[27] J. Nivre, J. Hall, J. Nilsson, A. Chanev, G. Eryigit, S. Kübler, S. Marinov, E. Marsi, “MaltParser: A language-independent system for data-driven dependency parsing,” In the Journal of Natural Language Engineering, Volume. 13, No. 2, pp. 95-135, June 2007.

[28] J. Nivre, J. Hall, S. Kübler, R. McDonald, J. Nilsson, S. Riedel, and D. Yuret, “The CoNLL 2007 shared task on dependency parsing,” In the Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL), pp. 915-932, June 2007.

[29] J. Tiedemann, and Z. Agić, “Synthetic treebanking for cross-lingual dependency parsing,” In the Journal of Artificial Intelligence Research, Volume. 55, pp. 209-248, January 2016. [3-r-5]

[30] John Lafferty, Andrew McCallum, and Fernando C. N. Pereira, “Conditional random fields: Probabilistic models for segmenting and labeling sequence data,” In the Proceedings of the 18th International Conference on Machine Learning (ICML), pp. 282-289, 2001.

[31] K. Greff, RK. Srivastava, J. Koutník, BR. Steunebrink, and J. Schmidhuber, “A search space odyssey,” In the Journal of IEEE transactions on neural networks and learning systems, Volume. 28, No. 10, pp. 2222-2232, July 2016.

[32] K. Haverinen, J. Nyblom, T. Viljanen, V. Laippala, S. Kohonen, A.

Missilä, S. Ojala, T. Salakoski, and F. Ginter, “Building the essential resources for Finnish: the Turku Dependency Treebank,” In the Journal of Language Resources and Evaluation, Volume. 48, No. 3, pp. 493-531, September 2014.

[33] K. Haverinen, J. Nyblom, T. Viljanen, V. Laippala, S. Kohonen, A. Missilä , S. Ojala, T. Salakoski, F. Ginter, “Building the essential resources for Finnish: the Turku Dependency Treebank,” In the Journal of Language Resources and Evaluation. 2014, Volume. 48, No. 3, pp. 493-531, September 2014.

[34] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, “BLEU: a method for automatic evaluation of machine translation,” In the Proceedings of the 40th Annual Meeting on Association for Computational Linguistics Published by Association for Computational Linguistics , pp. 311-318, July 2002.

[35] K. W. W. Htike, Y. K. Thu, Z. Zhang, W. P. Pa, Y. Sagisaka, N. Iwahashi, “Comparison of Six POS Tagging Methods on 10K Sentences Myanmar Language (Burmese) POS Tagged Corpus,” In the Proceedings of 18th International Conference on Intelligent Text Processing and Computational Linguistics (CICLING), April 2017.

[36] L. Huang, W. Jiang, and Q. Liu. “Bilingually-constrained (monolingual) shift-reduce parsing,” In the Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3-Volume 3, pp. 1222-1231, August 2009.

[37] L. Versteegen, “The simple Bayesian classifier as a classification algorithm,” 1999 [Online] Available: http://www. cs. kun. nl/nsccs/artikelen/leonv. ps. Z. 2000.

[38] M. Seraji, C. Jahani, B. Megyesi, and J. Nivre, “A Persian treebank with Stanford typed dependencies,” In the Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC), pp. 796-801, May 2014.

[39] M. Straka, J. Hajic, J. Strakov ˇ a, “UDPipe: Trainable Pipeline for Processing CoNLL-U Files Performing Tokenization, Morphological Analysis, POS Tagging and Parsing,” In the

Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC 2016), pp. 1-10, May 2016.

[40] MC. De Marneffe, and CD. Manning, "The Stanford typed dependencies representation," In Coling 2008: Proceedings of the workshop on cross-framework and cross-domain parser evaluation, pp. 1-8. Association for Computational Linguistics, August 2008.

[41] MC. De Marneffe, B. MacCartney, and CD. Manning, "Generating typed dependency parses from phrase structure parses," In the Proceedings of LREC 2006, Volume 6, pp. 449-454, May 2006.

[42] MC. de Marneffe, M. Connor, N. Silveira, SR. Bowman SR, T. Dozat, and CD. Manning, "More constructions, more genres: Extending Stanford dependencies," In the Proceedings of the Second International Conference on Dependency Linguistics (DepLing 2013), pp. 187-196, August 2013.

[43] MC. De Marneffe, T. Dozat, N. Silveira, K. Haverinen, F. Ginter, and J. Nivre, "Universal Stanford dependencies: A cross-linguistic typology," In the LREC 2014, Volume. 14, pp. 4585-4592, May 2014.

[44] MC. De Marneffe, T. Dozat, N. Silveira, K. Haverinen, F. Ginter, and J. Nivre, "Universal Stanford dependencies: A cross-linguistic typology," In the Proceedings of LREC 2014, p. 4585-4592, May 2014.

[45] N. Green, "Dependency parsing," In the Journal of WDS 2011 Proceedings of Contributed Papers, pp. 137–142, 2011.

[46] N. Green, SD. Larasati, and Z. Žabokrtský, "Indonesian dependency treebank: Annotation and parsing," In the Proceedings of the 26th Pacific Asia Conference on Language, Information, and Computation, pp. 137-145. Linköping University Electronic Press, November 2012.

[47] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overftting," In the Journal of machine learning research, Volume. 15, No. 1, pp. 1928-1958, January 2014.

[48] P. Koehn, F. J. Och, and D. Marcu, "Statistical phrase-based Translation," In the Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1, pp. 48-54. Association for Computational Linguistics, May 2003.

[49] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, and C. Dyer, "Moses: Open source toolkit for statistical machine translation," In the Proceedings of the 45th annual meeting of the association for computational linguistics (ACL) companion volume proceedings of the demo and poster sessions 2007 , pp. 177–180, June 2007.

[50] PC. Chang, H. Tseng, D. Jurafsky, and CD. Manning, "Discriminative reordering with Chinese grammatical relations features," In the Proceedings of the Third Workshop on Syntax and Structure in Statistical Translation, pp. 51-59. Association for Computational Linguistics, June 2009.

[51] PE. Solberg, "Building gold-standard treebanks for Norwegian," In the Proceedings of the 19th Nordic Conference of Computational Linguistics (NODALIDA 2013), No. 085, pp. 459-464. Linköping University Electronic Press, May 2013.

[52] R. Fujii, R. Domoto, and D. Mochihashi , "Nonparametric Bayesian semi-supervised word segmentation," In the Journal of Transactions of the Association for Computational Linguistics, Volume. 5, pp. 179-189, December 2017. [6-r-cf14]

[53] R. L. Gomes and E. R. M. Madeira, "Converting Russian dependency treebank to Stanford typed dependencies representation," In the Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, Volume 2: Short Papers, pp. 143-147, April 2014.

[54] R. McDonald, and J. Nivre, "Characterizing the errors of data-driven dependency parsing models," In the Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning

(EMNLP-CoNLL), 2007.

[55] R. Rosa, J. Masek, D. Marecek, M. Popel, D. Zeman, and Z. Zabokrtský, "HamleDT 2.0: Thirty Dependency Treebanks Stanfordized," In the Proceedings of LREC 2014, pp. 2334-2341, May 2014.

[56] R. Tsarfaty, "A unified morpho-syntactic scheme of Stanford dependencies," In the Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pp. 118-124, December 2002.

[57] S. Buchholz, and E. Marsi, "CoNLL-X shared task on multilingual dependency parsing," In the Proceedings of the tenth conference on computational natural language learning, pp. 149-164. Association for Computational Linguistics, June 2006.

[58] S. F. Chen, and J. Goodman, "An empirical study of smoothing techniques for language modeling," In the Journal of Computer Speech & Language, Volume. 13, No. 4, pp. 359-394, October 1999.

[59] S. Hochreiter, and J. Schmidhuber, "Long short-term memory," In the Journal of Neural Computation, Volume. 9, No. 8, pp. 1735-1780, November 1997.

[60] S. Kubler, R. McDonald, and J. Nivre, "Dependency parsing," In the Journal of Synthesis lectures on human language technologies, Volume.1, No.1, pp. 1-27, January 2009.

[61] S. Mori, H. Ogura, and T. Sasada, "A Japanese Word Dependency Corpus," In the Proceedings of LREC, pp. 753-758, May 2014.

[62] S. Petrov, D. Das, and R. McDonald, "A universal part-of-speech tagset," In the Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12), pp. 2089-2096. European Language Resources Association (ELRA), May 2012.
[2-ur-up]

[63] S. Petrov, D. Das, and R. McDonald, "A Universal Part-of-Speech Tagset," In the Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12) , pp. 2089–2096. European Language Resources Association (ELRA), May

2012.

[64] T. Kakkonen, “Dependency treebanks: Methods, annotation schemes and tools,” In the Proceedings of the 15th Nordic Conference of Computational Linguistics (NODALIDA 2005), pp. 94-104, 2005.

[65] T. Kudo and Y. Matsumoto, “Japanese Dependency Analysis Using Cascaded Chunking,” In the Proceedings of the 6th conference on Natural language learning-Volume 20 Published by Association for Computational Linguistics, pp. 1-7, August 2002.

[66] T. Tanaka, Y. Miyao, M. Asahara, S. Uematsu, H. Kanayama, S. Mori, Y. Matsumoto, “Universal dependencies for Japanese,” In the Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16), pp. 1651-1658, May 2016.

[67] W. W. Thant, T. M. Htwe, N. Thein, “Context Free Grammar Based Top-Down Parsing of Myanmar Sentences,” In the Proceedings of International conference on computer science and information technology, pp. 71-75, December 2011.

[68] W. W. Thant, T. M. Htwe, N. Thein, “Parsing of MYANMAR Sentences with Function Tagging,” International Journal of Computer Applications, Volume 26, No. 2, July 2011.

[69] X. Glorot, and Y. Bengio, “Understanding the difficulty of training deep feedforward neural networks,” In Proceedings of the thirteenth International Conference on Artificial Intelligence and Statistics AISTATS (PMLR), pp. 249-256, March 2010.

[70] Y. K. Thu, W. P. Pa M. Utiyama, A. Finch, E. Sumita, “Introducing the Asian Language Treebank (ALT),” In the Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16), pp. 1574 - 1578, May 2016.

[71] Y. Zhang, and J. Nivre, “Transition-based dependency parsing with rich non-local features,” In the Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2, pp.188-193. Association for Computational Linguistics, June 2011.

APPENDICES

Appendix A: Development of Myanmar Dependency Parsing

There are two main parts to develop Myanmar dependency parsing system. The first one is training part which is building Myanmar dependency parsing model by applying transition based dependency parsing technique based on predictions of transitions by neural networks classifier. The second part is testing the performance of the trained dependency parsing model. A simulated dependency parsing system has been built to test the trained model. The detail procedures of developing Myanmar dependency parsing will be described in this appendix.

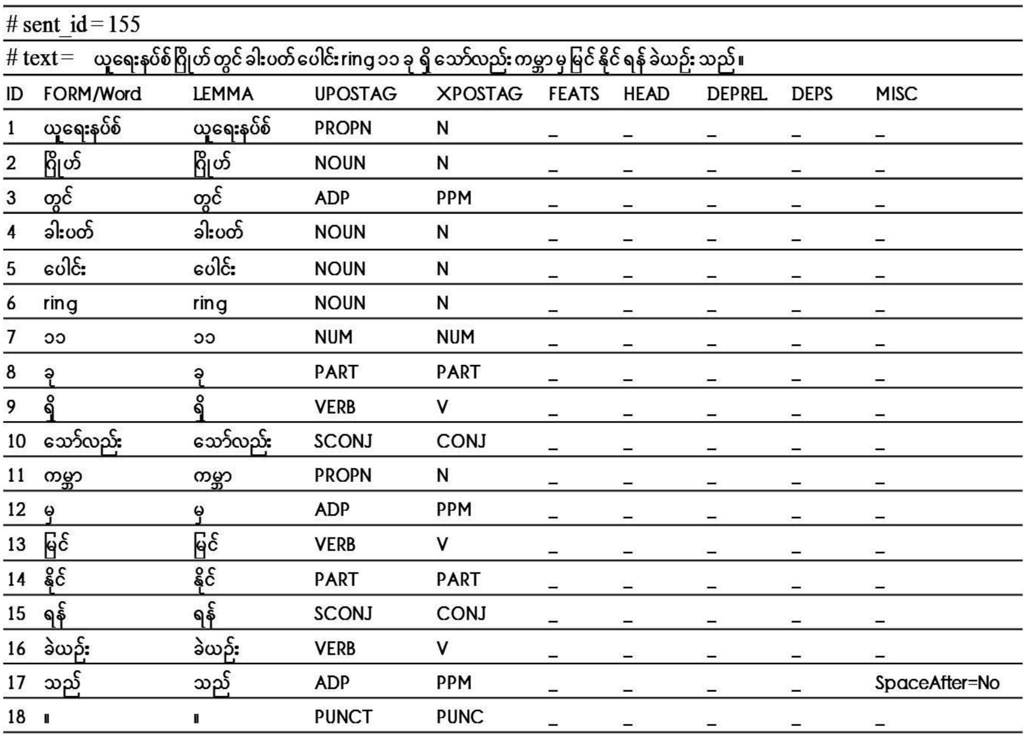
1. Data Preprocessing of Myanmar Dependency Treebank

In data preprocessing, the first step is checking word segmentation and part-of-speech (POS) tagging of selected corpora for treebank to be unique format for the whole treebank as the corpora were segmented and POS tagged based on different creation purposes and unique word segmentation and POS tagging is easy to add related Universal part-of-speech. Moreover, unique format in word segmentation and POS tagging is able to make reliable and fast tagging related Universal part-of-speech. Data preprocessing step was carried out by three steps described in the following sub titles.

1.1 Adding Universal POS Tags

Most Myanmar POS tags are very similar to Universal POS tags. Therefore, those similar POS tags of selected corpora were added by python program scripts. The other different POS tags like conjunction, CONJ, were checked and added right related Universal POS tags manually.

The corpora were prepared as Universal Dependencies frame (CoNLL-U) format with Myanmar language POS tags and Universal POS tags after data preprocessing step. The example sentence of corpora can be seen as the following figure.



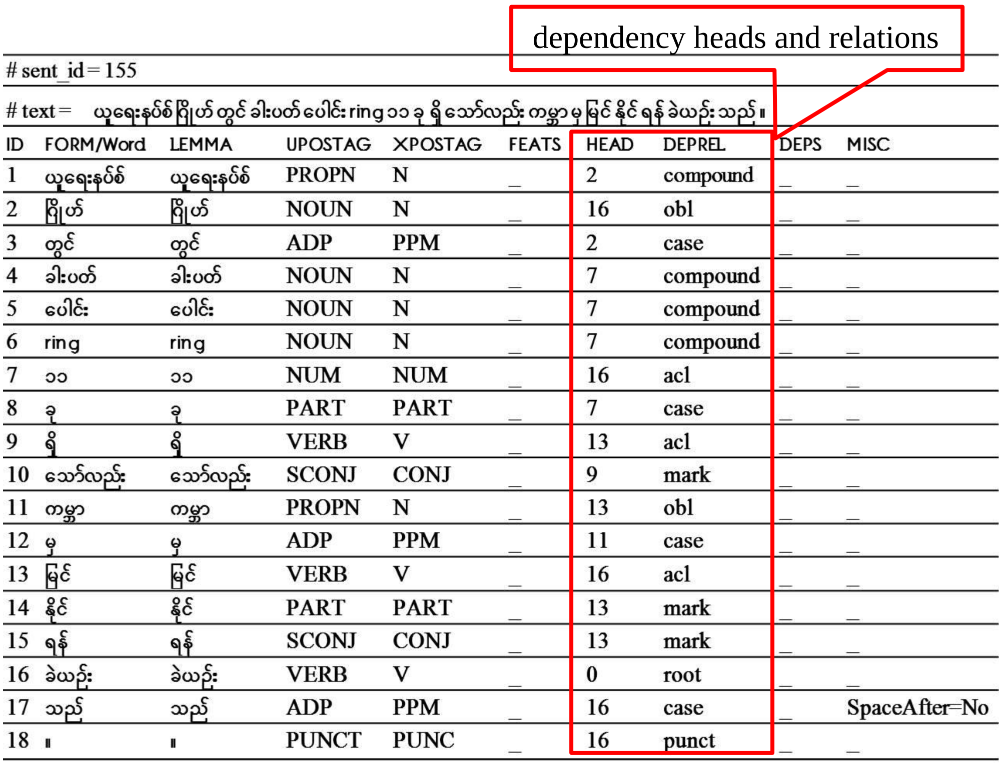
# sent_id = 155									
# text =	ယူရေးနပ်စ် ဂြိုဟ် တွင် ခါးပတ် ပေါင်း ring ၁၁ ခု ရှိ သော်လည်း ကမ္ဘာ မှ မြင် နိုင် ရန် ခဲယဉ်း သည် ။								
ID	FORM/Word	LEMMA	UPOSTAG	XPOSTAG	FEATS	HEAD	DEPREL	DEPS	MISC
1	ယူရေးနပ်စ်	ယူရေးနပ်စ်	PROPN	N	_	_	_	_	_
2	ဂြိုဟ်	ဂြိုဟ်	NOUN	N	_	_	_	_	_
3	တွင်	တွင်	ADP	PPM	_	_	_	_	_
4	ခါးပတ်	ခါးပတ်	NOUN	N	_	_	_	_	_
5	ပေါင်း	ပေါင်း	NOUN	N	_	_	_	_	_
6	ring	ring	NOUN	N	_	_	_	_	_
7	၁၁	၁၁	NUM	NUM	_	_	_	_	_
8	ခု	ခု	PART	PART	_	_	_	_	_
9	ရှိ	ရှိ	VERB	V	_	_	_	_	_
10	သော်လည်း	သော်လည်း	SCONJ	CONJ	_	_	_	_	_
11	ကမ္ဘာ	ကမ္ဘာ	PROPN	N	_	_	_	_	_
12	မှ	မှ	ADP	PPM	_	_	_	_	_
13	မြင်	မြင်	VERB	V	_	_	_	_	_
14	နိုင်	နိုင်	PART	PART	_	_	_	_	_
15	ရန်	ရန်	SCONJ	CONJ	_	_	_	_	_
16	ခဲယဉ်း	ခဲယဉ်း	VERB	V	_	_	_	_	_
17	သည်	သည်	ADP	PPM	_	_	_	_	SpaceAfter=No
18	။	။	PUNCT	PUNC	_	_	_	_	_

1.2 Adding Raw Dependency Information by Unsupervised Annotation

Any dependency resource had not developed for Myanmar language and some grammar structures are very similar in Japanese and Myanmar language. Therefore, the first preprocessed corpus, myPOS, was added raw dependency structures using unsupervised dependency parsing by UDPipe via the Japanese model shared at UD project link: https://lindat.mff.cuni.cz/repository/xmlui/handle/11234/1-2364. The first corpus prepared with the CoNLL-U format was annotated by unsupervised approach by means of the following command.

```
./udpipe --parse  /japanese-ud-2.0-170801.udpipe  [UTF-8_encoded_input_file] >
[output_file]
```

The output file is in the CoNLL-U format and contains the dependency heads and dependency relations of sentences. The dependency head and relation of example sentence of the output file can be seen in the following figure.



dependency heads and relations

sent_id = 155

text = ယူရေးနပ်စ် ဂြိုဟ် တွင် ခါးပတ် ပေါင်း ring ၁၁ ခု ရှိ သော်လည်း ကမ္ဘာ မှ မြင် နိုင် ရန် ခဲယဉ်း သည် ။

ID	FORM/Word	LEMMA	UPOSTAG	XPOSTAG	FEATS	HEAD	DEPREL	DEPS	MISC
1	ယူရေးနပ်စ်	ယူရေးနပ်စ်	PROPN	N	_	2	compound	_	_
2	ဂြိုဟ်	ဂြိုဟ်	NOUN	N	_	16	obl	_	_
3	တွင်	တွင်	ADP	PPM	_	2	case	_	_
4	ခါးပတ်	ခါးပတ်	NOUN	N	_	7	compound	_	_
5	ပေါင်း	ပေါင်း	NOUN	N	_	7	compound	_	_
6	ring	ring	NOUN	N	_	7	compound	_	_
7	၁၁	၁၁	NUM	NUM	_	16	acl	_	_
8	ခု	ခု	PART	PART	_	7	case	_	_
9	ရှိ	ရှိ	VERB	V	_	13	acl	_	_
10	သော်လည်း	သော်လည်း	SCONJ	CONJ	_	9	mark	_	_
11	ကမ္ဘာ	ကမ္ဘာ	PROPN	N	_	13	obl	_	_
12	မှ	မှ	ADP	PPM	_	11	case	_	_
13	မြင်	မြင်	VERB	V	_	16	acl	_	_
14	နိုင်	နိုင်	PART	PART	_	13	mark	_	_
15	ရန်	ရန်	SCONJ	CONJ	_	13	mark	_	_
16	ခဲယဉ်း	ခဲယဉ်း	VERB	V	_	0	root	_	_
17	သည်	သည်	ADP	PPM	_	16	case	_	SpaceAfter=No
18	။	။	PUNCT	PUNC	_	16	punct	_	_

1.3 Post Processing Unsupervised Annotation Results

The first unsupervised annotated corpus, myPOS, was checked and updated manually according to the reference dependency structures in post processing step in order to be better and more reliable dependency structures in sentences. To be speedy post processing, selected 2,000 sentences from unsupervised annotated corpus were checked and updated by reference dependency structures and they were added to the training data to update the model and parse all sentences not updated in corpus. Then the post processed corpus was trained again through the following command.

```
cat [input_updated_file] | ./udpipe --tagger=none --tokenizer=none --train
[training_opts] udpipe_model
```

Then all the rest sentences not updated in corpus were parsed again by the updated model. Selected sentences from parsed results were checked and updated repeatedly until all sentences had been updated. After post processing first corpus, the first Myanmar parsing model was built by updated post processed data by UDPipe training command. The updated post processed model provided better annotated results to be faster manual post processing and it was used to support rapid annotation for other corpora: Web News and ALT.

After post processing and updating first Myanmar parsing model, Web News and ALT corpus are annotated by the post processed model. Then automatically annotated results of Web News data are checked and updated manually. After adding manual post processed data of myPOS and Web News corpora and unsupervised annotated data of ALT corpus were added to Myanmar dependency treebank and they were trained together to build Myanmar dependency parsing model.

Appendix B: Experimental Setup for UDPipe

The development processes of Myanmar dependency parsing and parsing experiments were implemented on CentOS 7 Linux virtual machine installed in the Intel® Core™ i7-5500U laptop. Download UDPipe at UDPipe development repository hosted on GitHub link: https://github.com/ufal/udpipe/. It is needed to install following software installation steps for UDPipe installation.

```
`- g++ 4.7 or newer, clang 3.2 or newer, Visual C++ 2015 or newer
`- make
`- SWIG 3.0.8 or newer for language bindings other than C++
```

Appendix C. Building Myanmar Dependency Parsing System

After post processing, Myanmar Dependency parsing system had been implemented as a web-based application by Flask package on CentOS 7 in a Python virtual environment. Details on how to install Flask in a Python virtual environment on CentOS 7 has been described at https://linuxize.com/post/how-to-install-flask-on-centos-7/ . In this system, "Myan-word-breaker" is used for word segmentation tool for input Myanmar sentences by slightly modifying main word segmentation python code to get raw word tokens and it can be downloaded at https://github.com/stevenay/myan-word-breaker. All data of Myanmar dependency treebank are used as training data for building tagger and parser model of this system by UDPipe. UDpipe command for training model of tagged model is as follow:

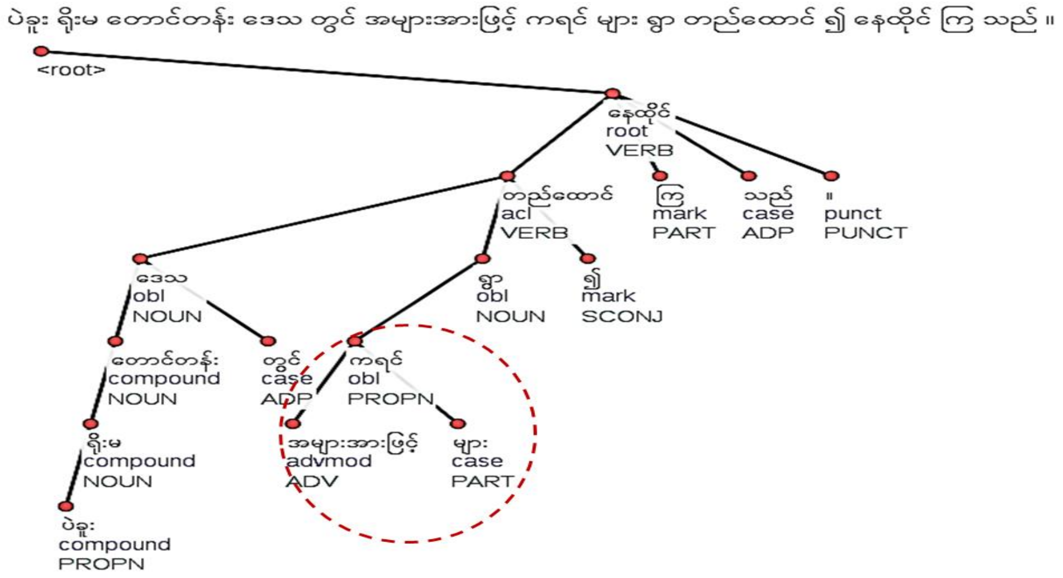
```
cat [input_updated_file] | ./udpipe --tokenizer=none --parser=none  --train
[training_opts] udpipe_tagger_model
```

To view output dependency trees fast, in this system, CoNLL-U Viewer is used and it can be downloaded at https://universaldependencies.org/tools.html#conll-u-viewer .

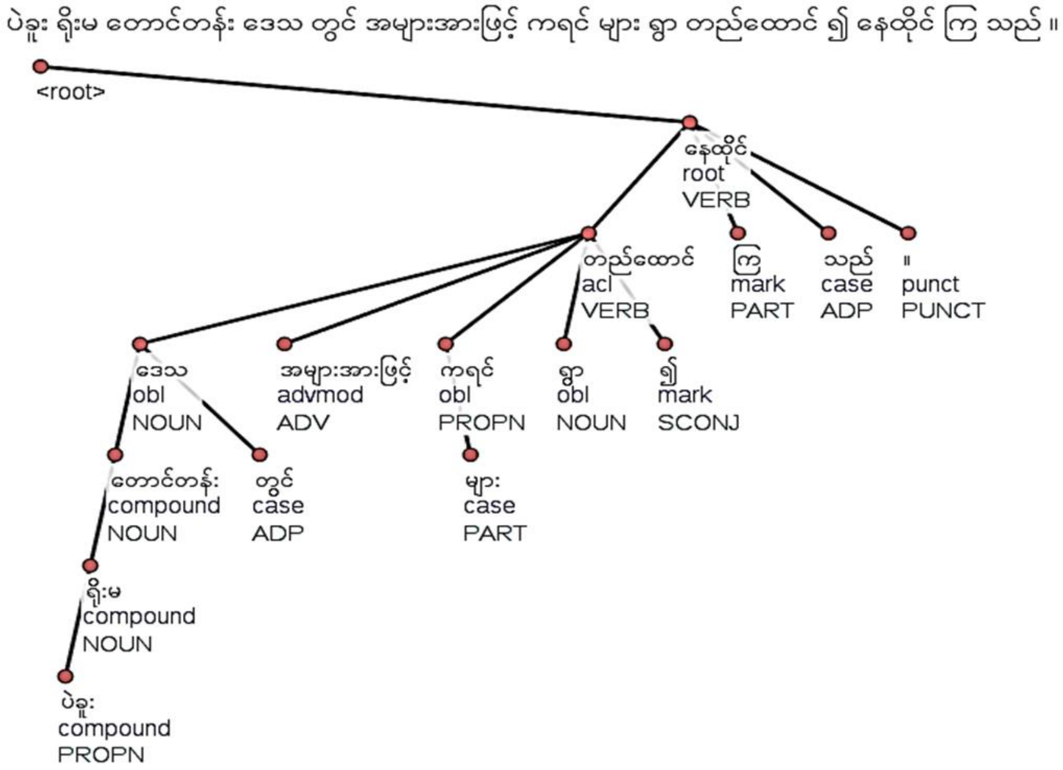
Appendix D: Example outputs of Myanmar Dependency Parsing Model

Example output trees of Myanmar dependency parsing system by current Myanmar dependency parsing model for open input test sentences are described in following tables. Errors in dependency head nodes in system output trees are circled.

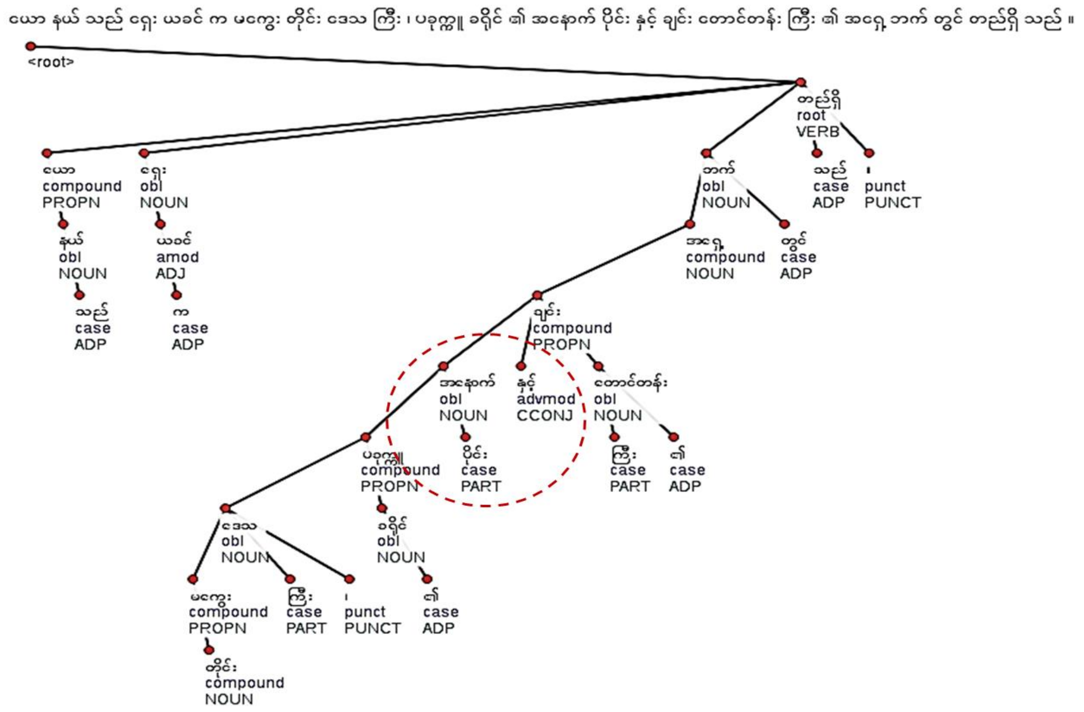
System model generated output



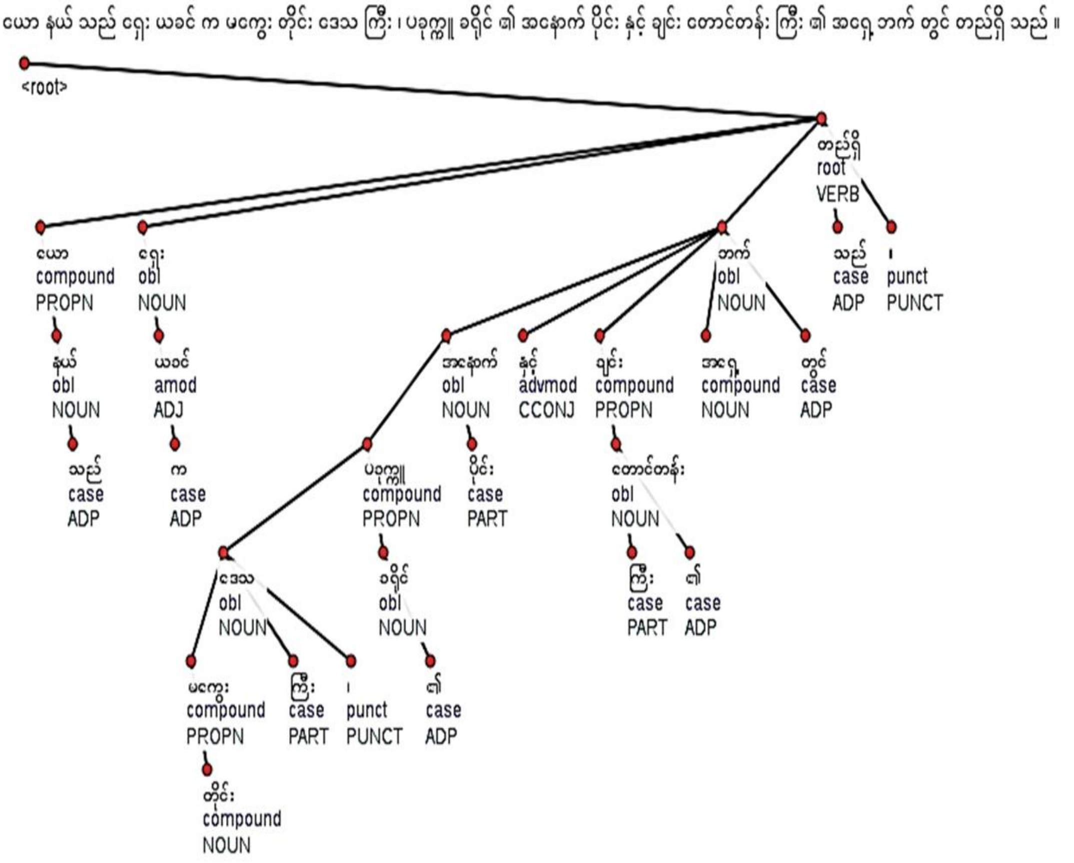
Reference Tree



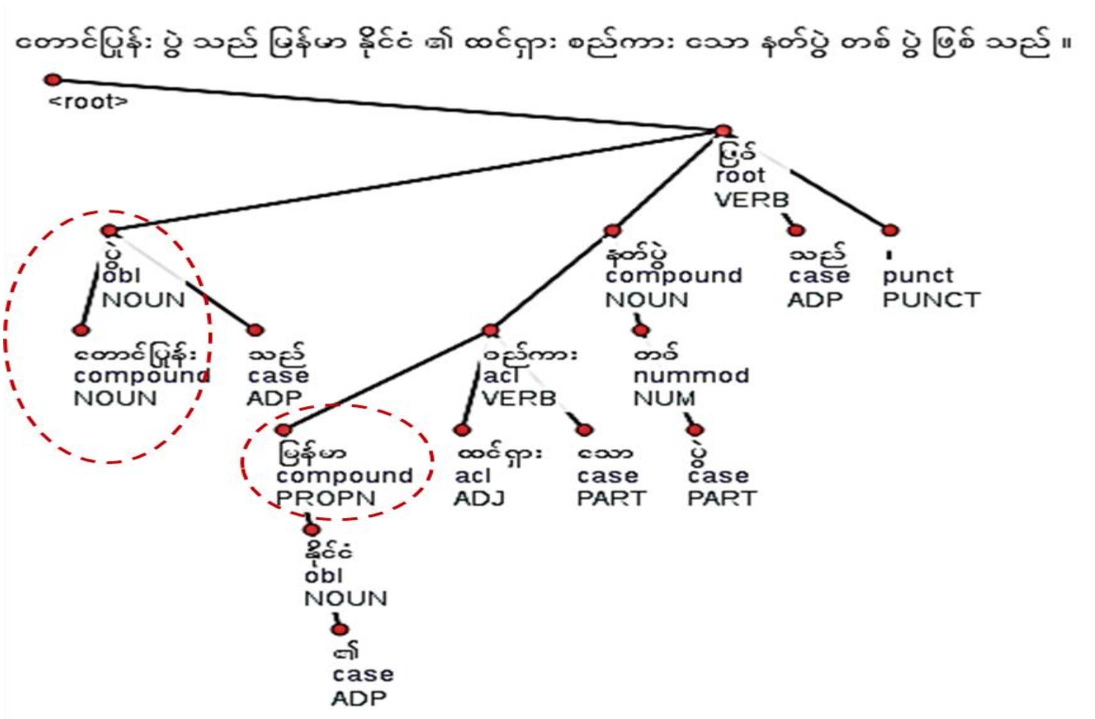
System model generated output



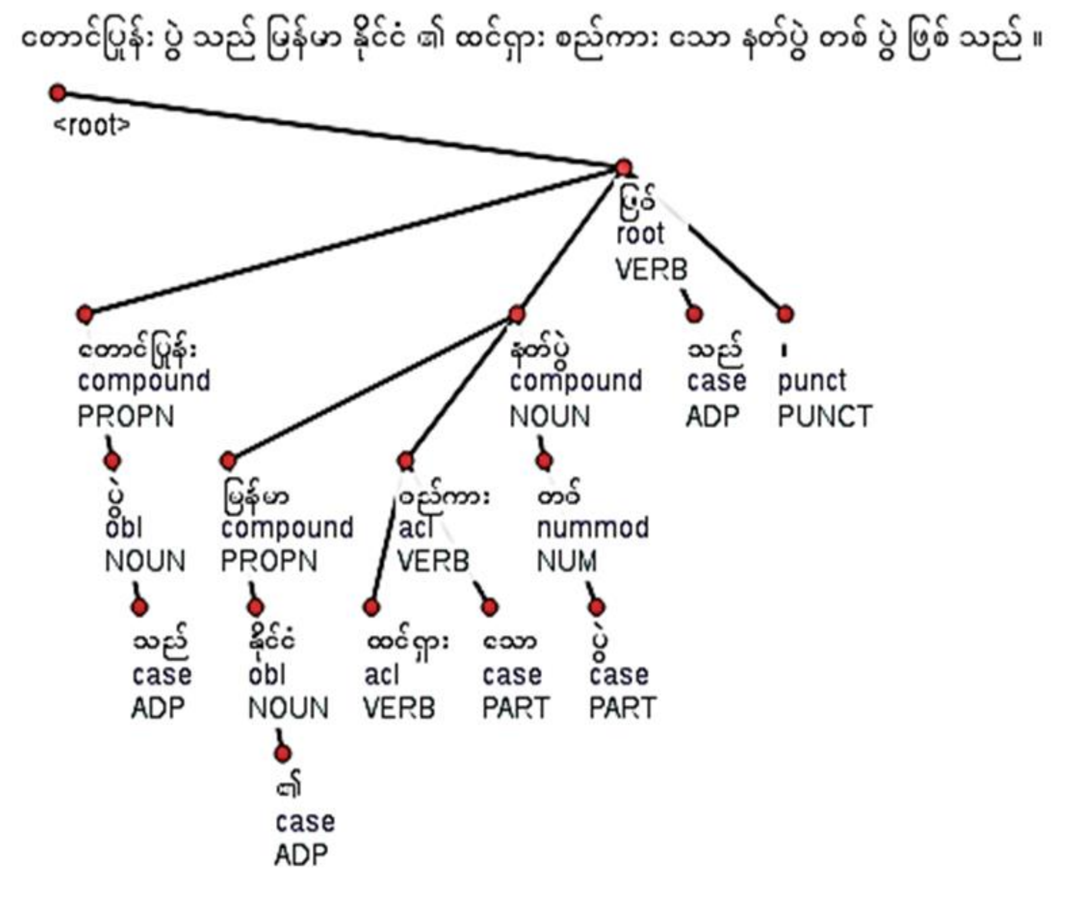
Reference Tree



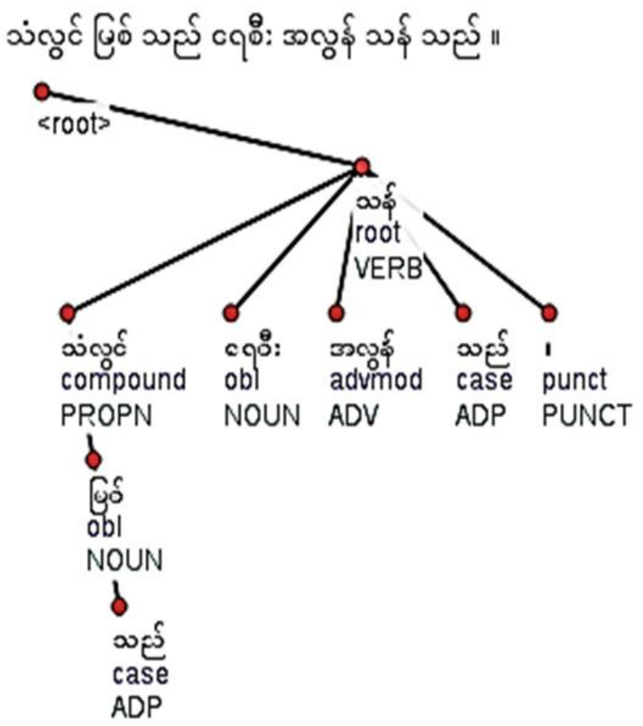
System model generated output



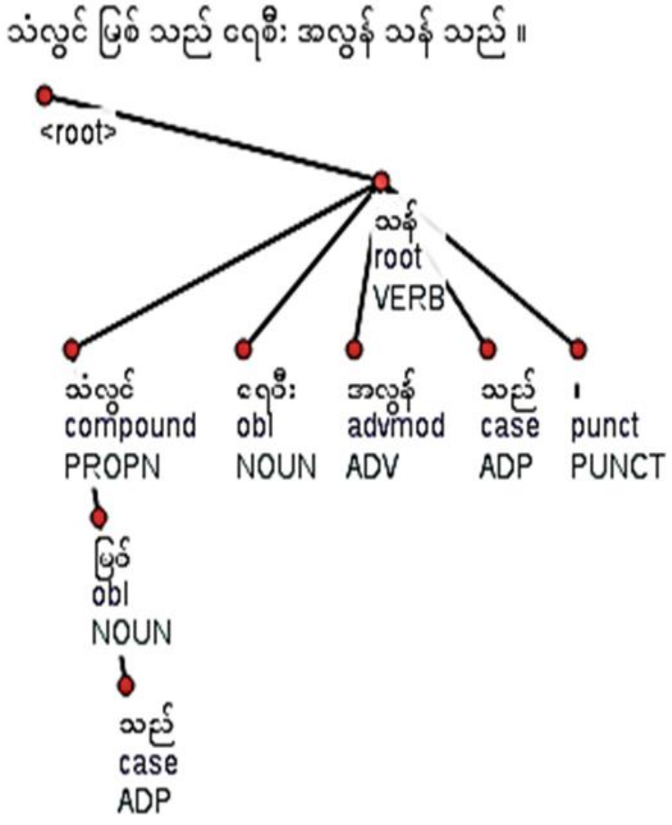
Reference Tree



System model generated output

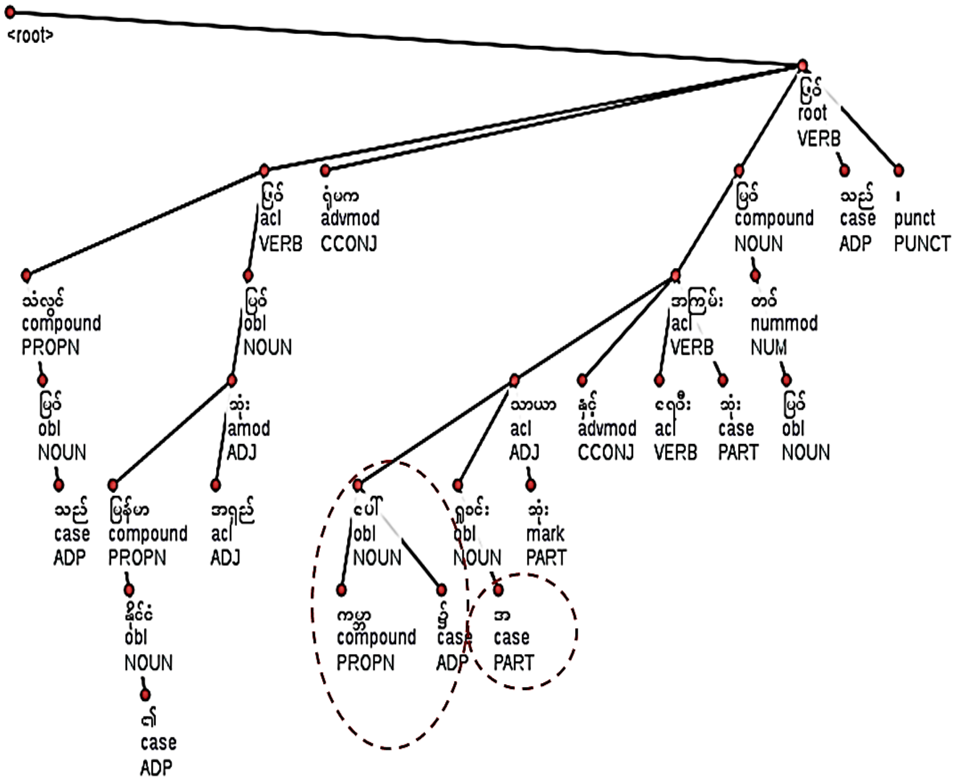


Reference Tree



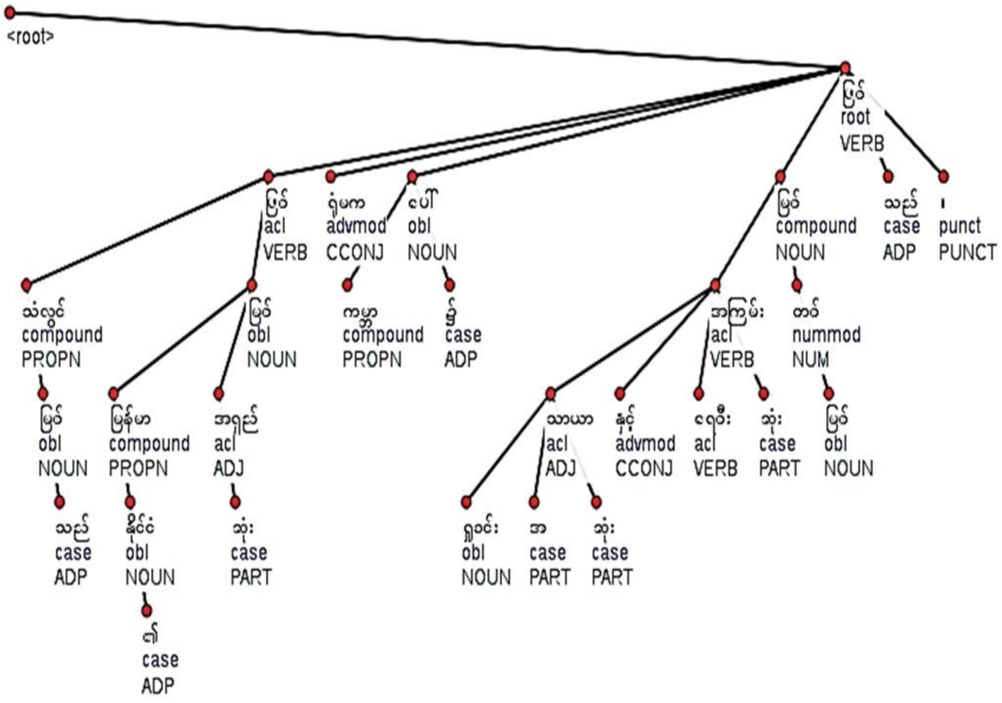
System model generated output

သံလွင် မြစ် သည် မြန်မာ နိုင်ငံ ၏ အရှည် ဆုံး မြစ် ဖြစ် ရုံမက ကမ္ဘာ ပေါ် ၌ ရှုခင်း အ သာယာ ဆုံး နှင့် ရေစီး အကြမ်း ဆုံး မြစ် တစ် မြစ် ဖြစ် သည် ။

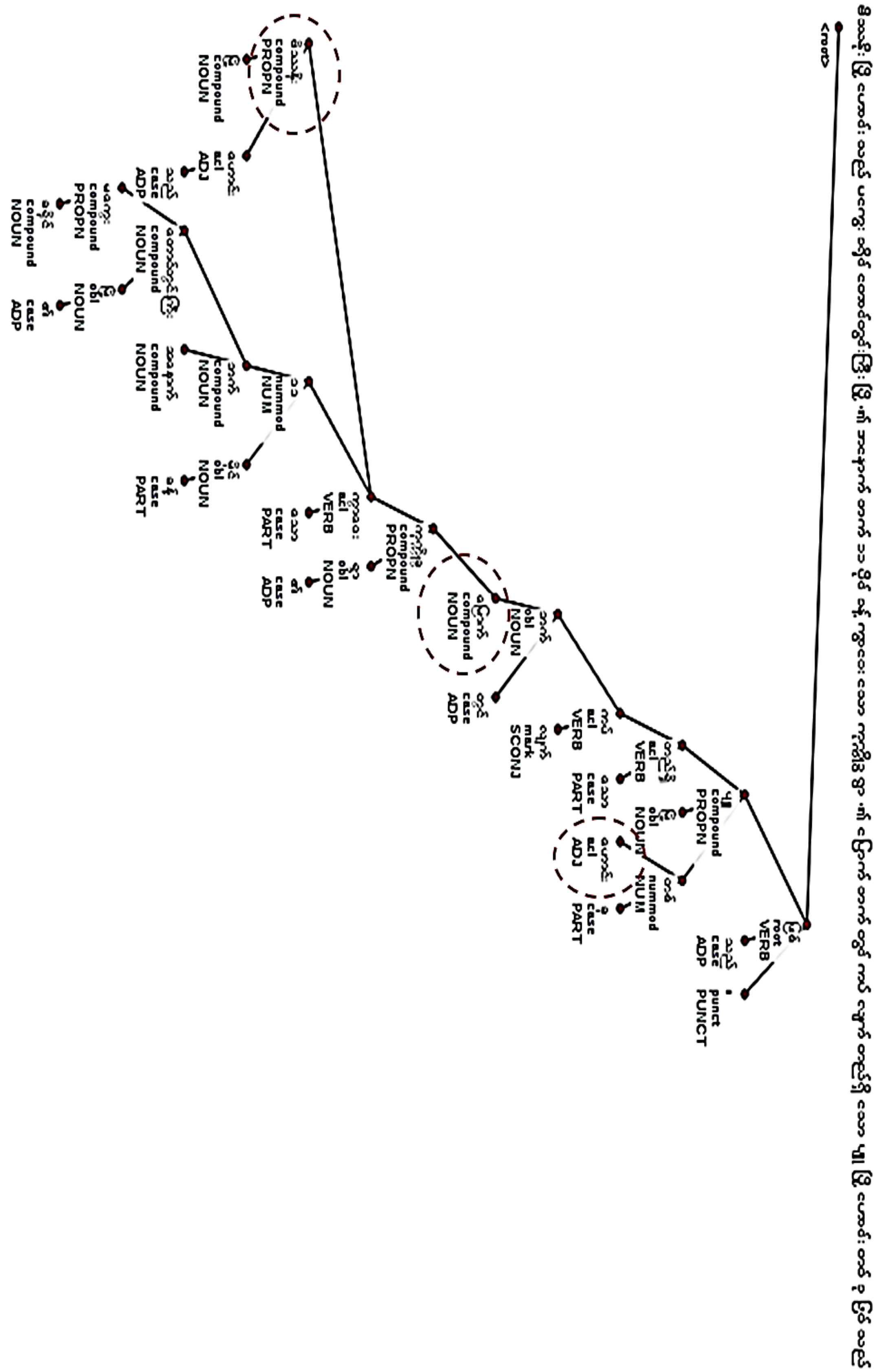


Reference Tree

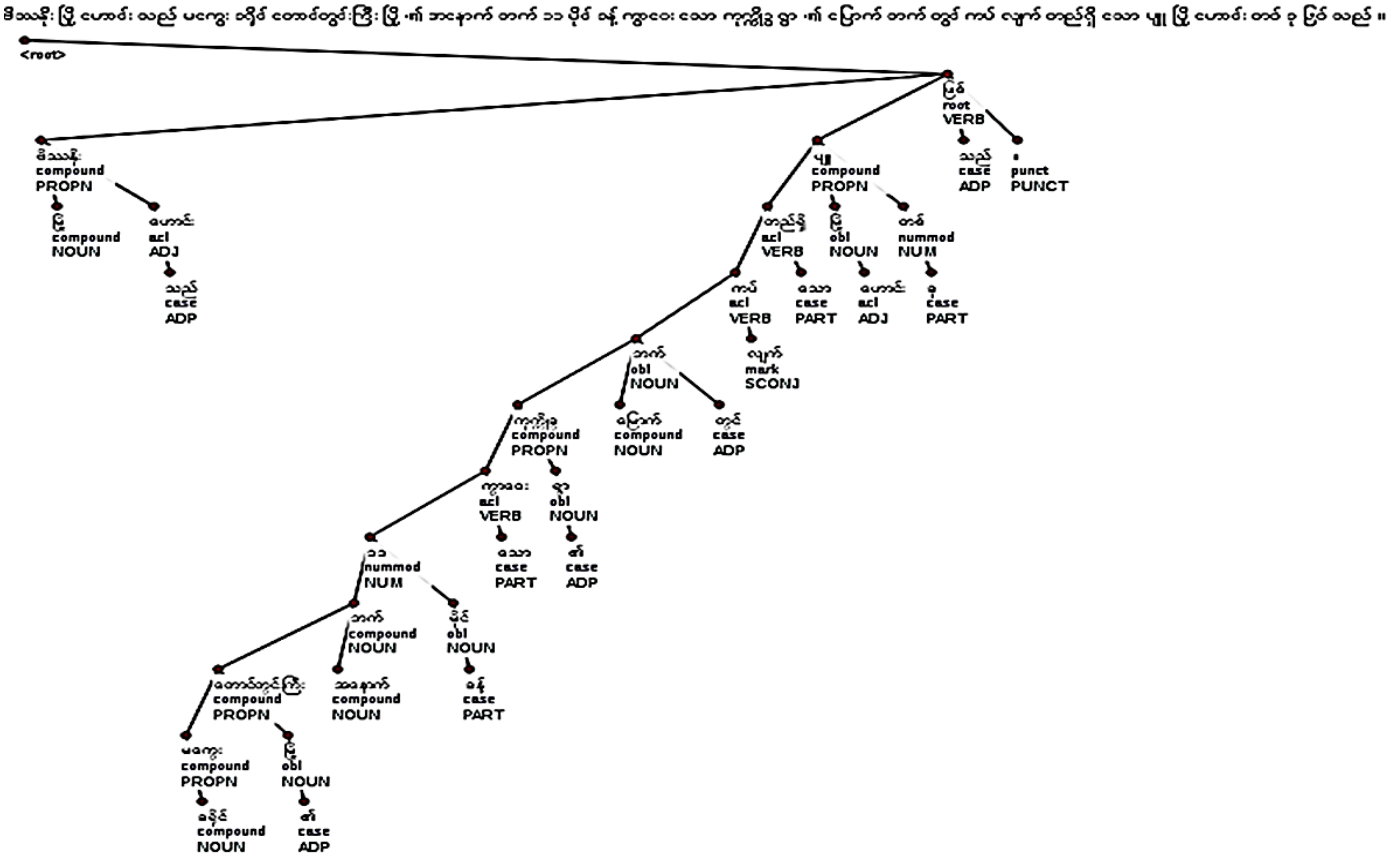
သံလွင် မြစ် သည် မြန်မာ နိုင်ငံ ၏ အရှည် ဆုံး မြစ် ဖြစ် ရုံမက ကမ္ဘာ ပေါ် ၌ ရှုခင်း အ သာယာ ဆုံး နှင့် ရေစီး အကြမ်း ဆုံး မြစ် တစ် မြစ် ဖြစ် သည် ။



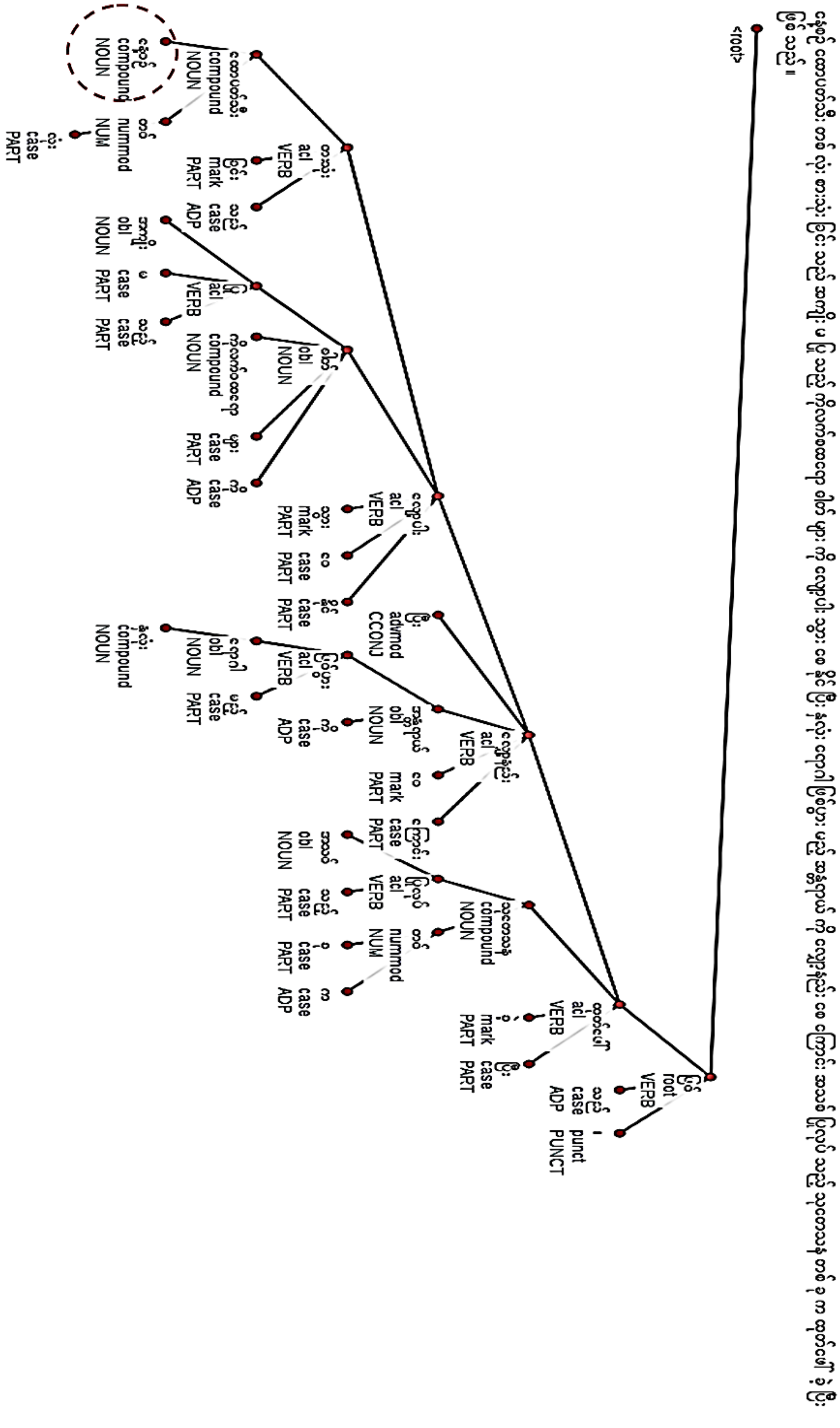
System model generated output



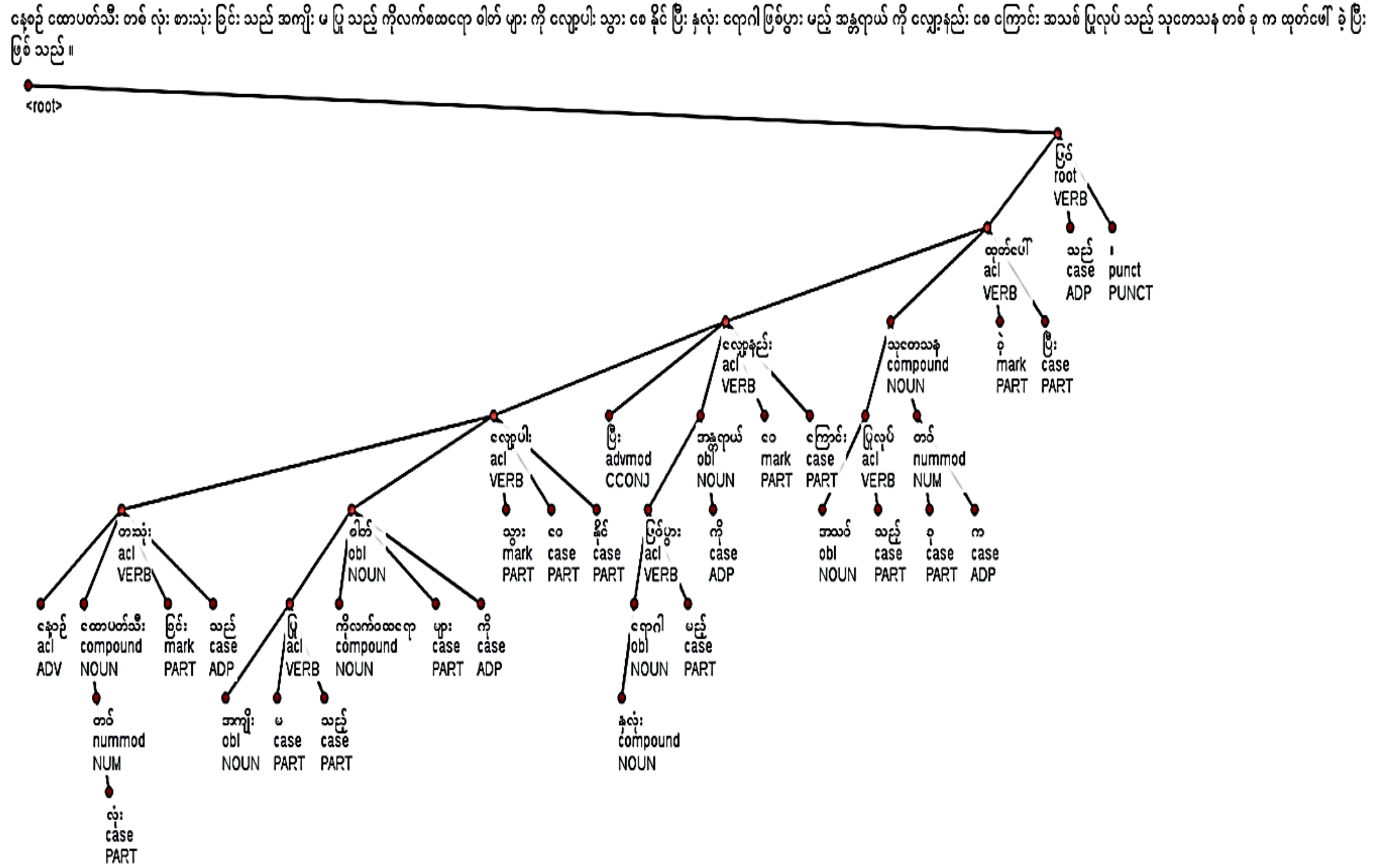
Reference Tree



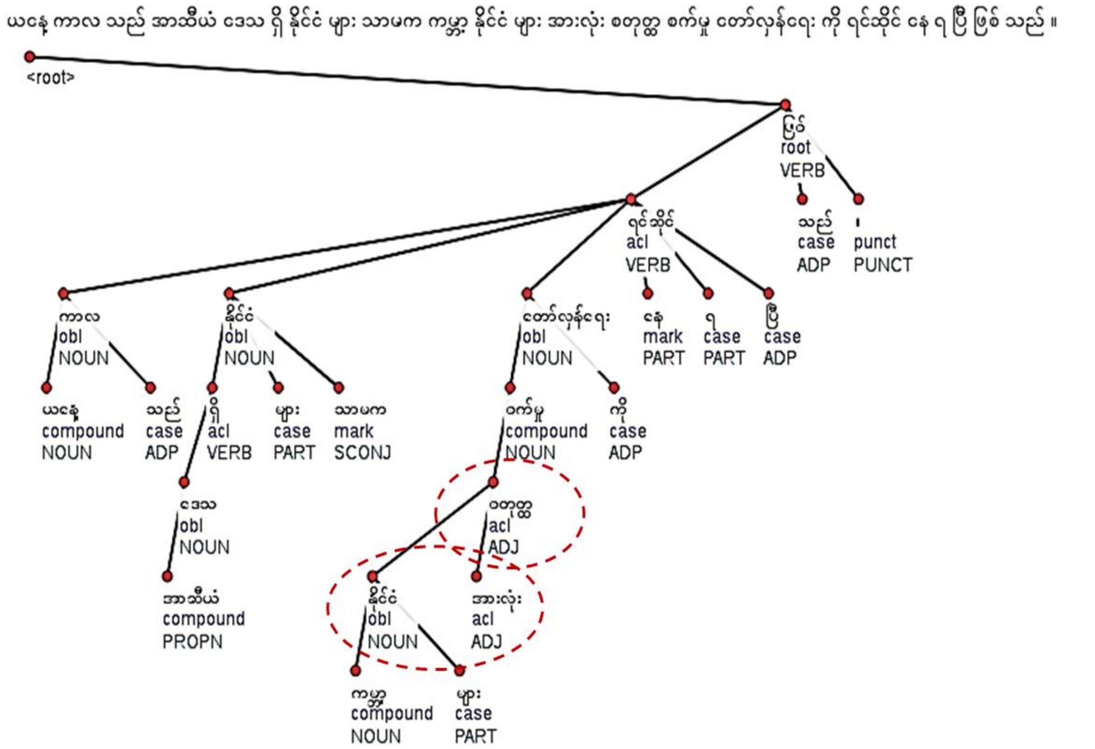
System model generated output



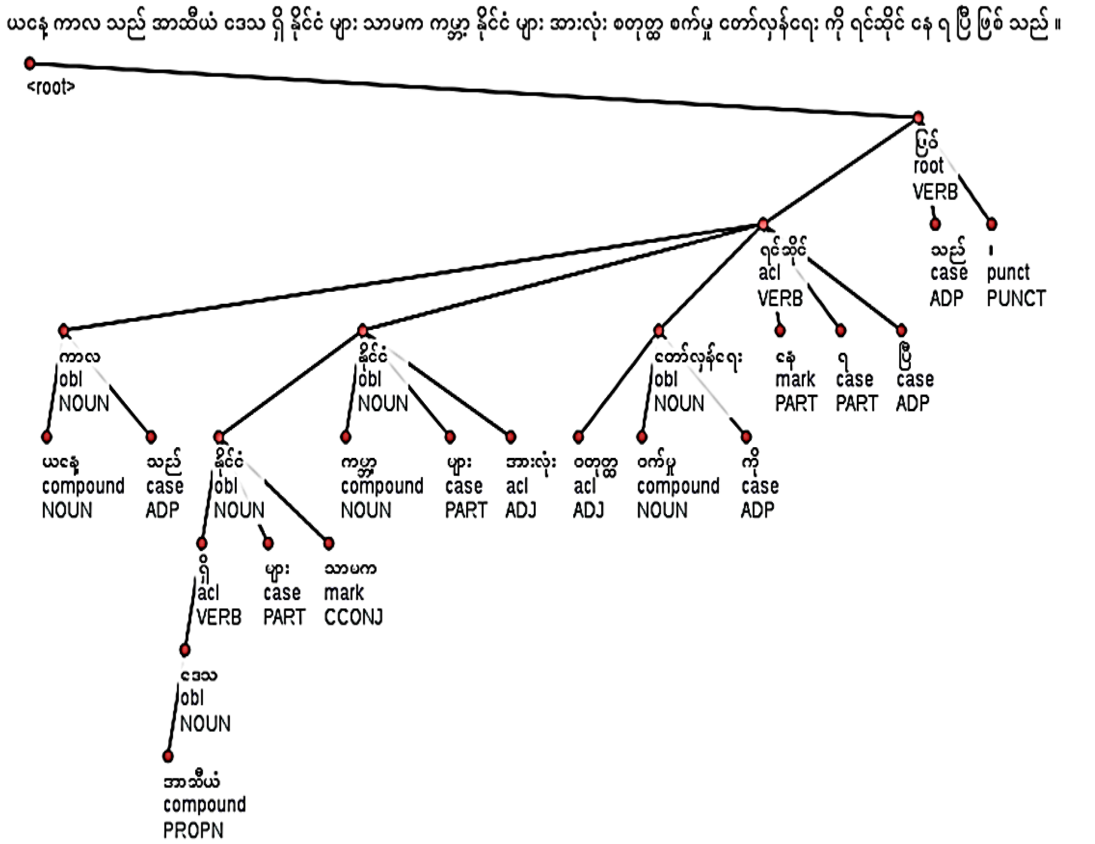
Reference Tree



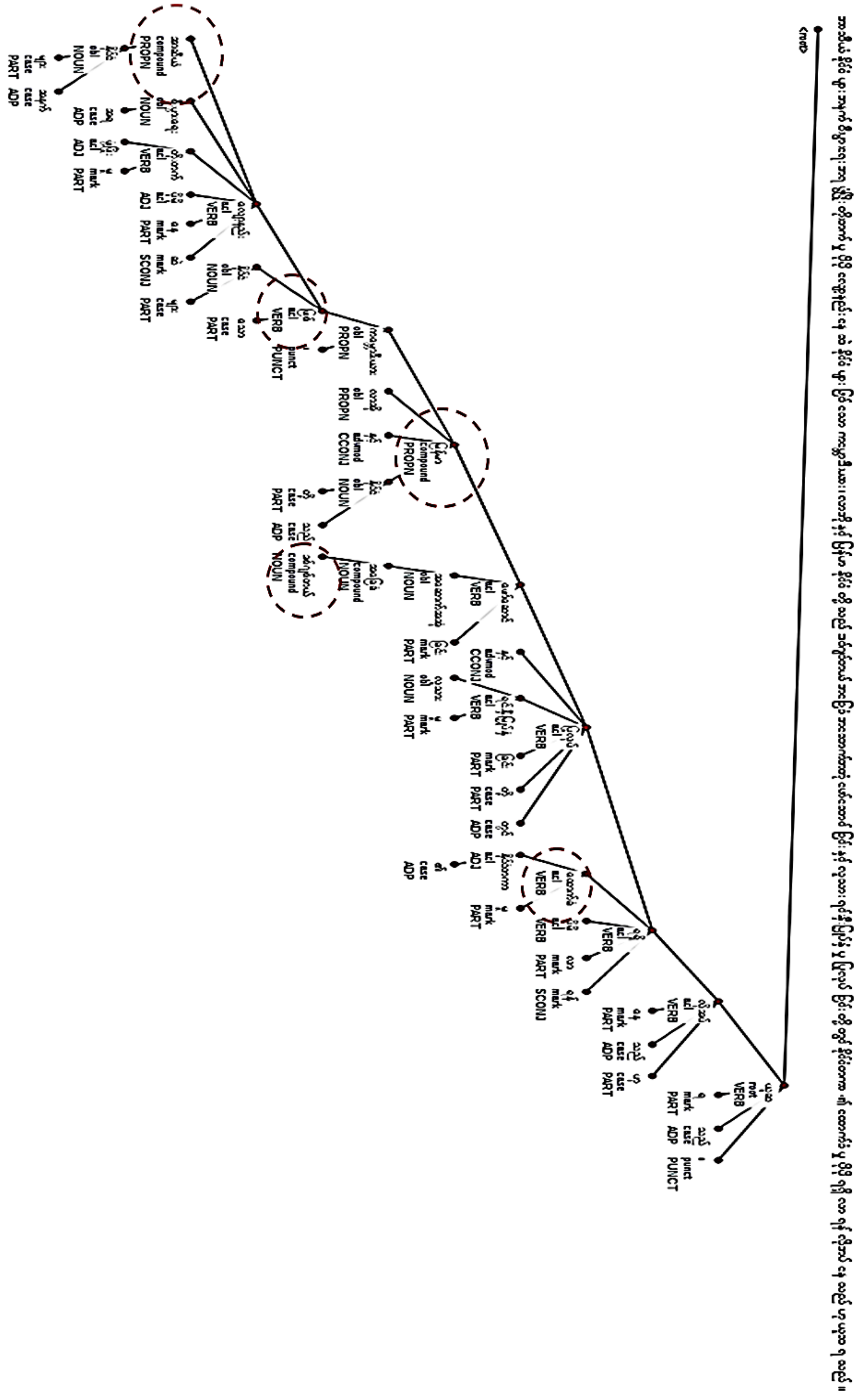
System model generated output



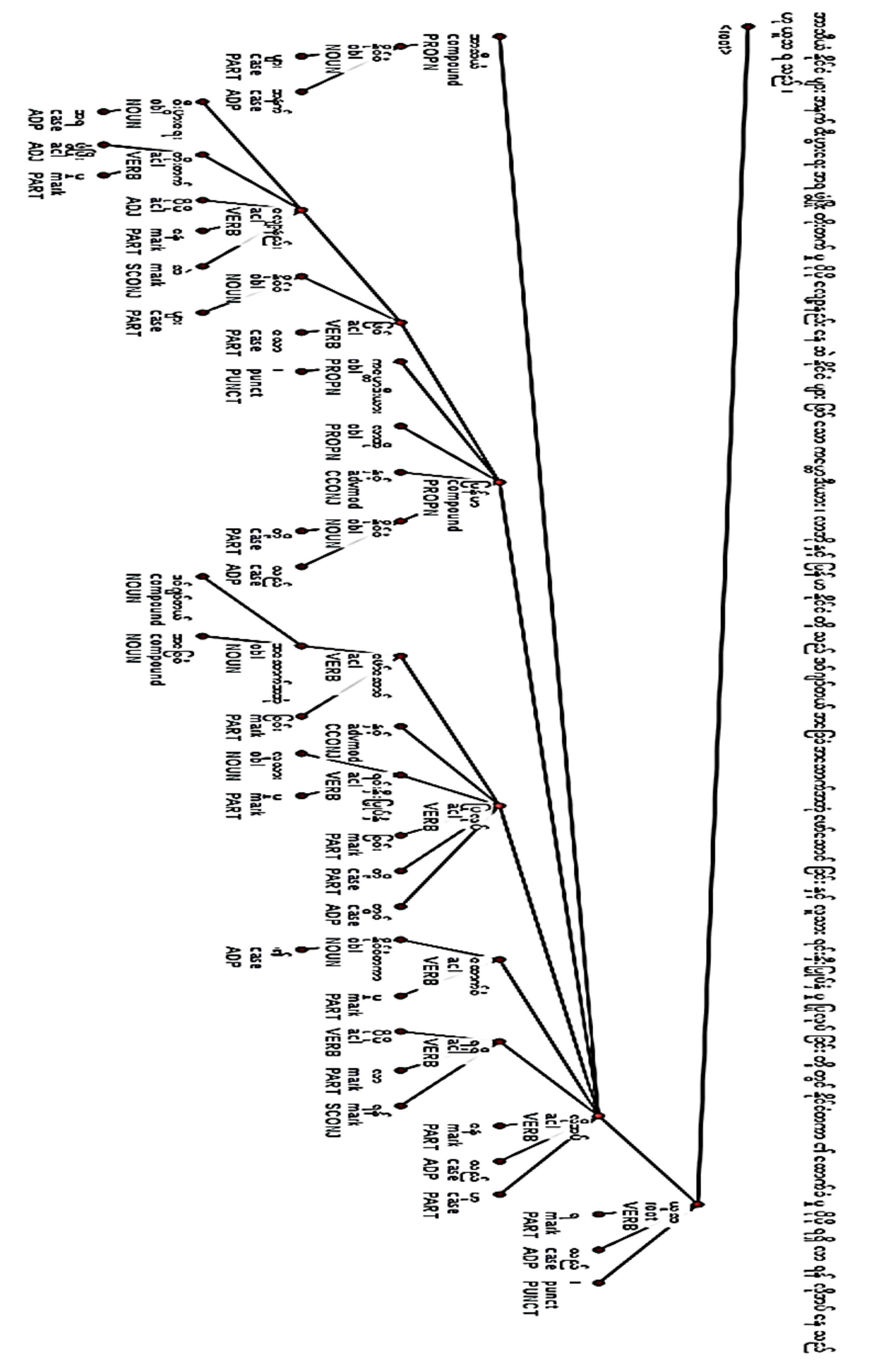
Reference Tree



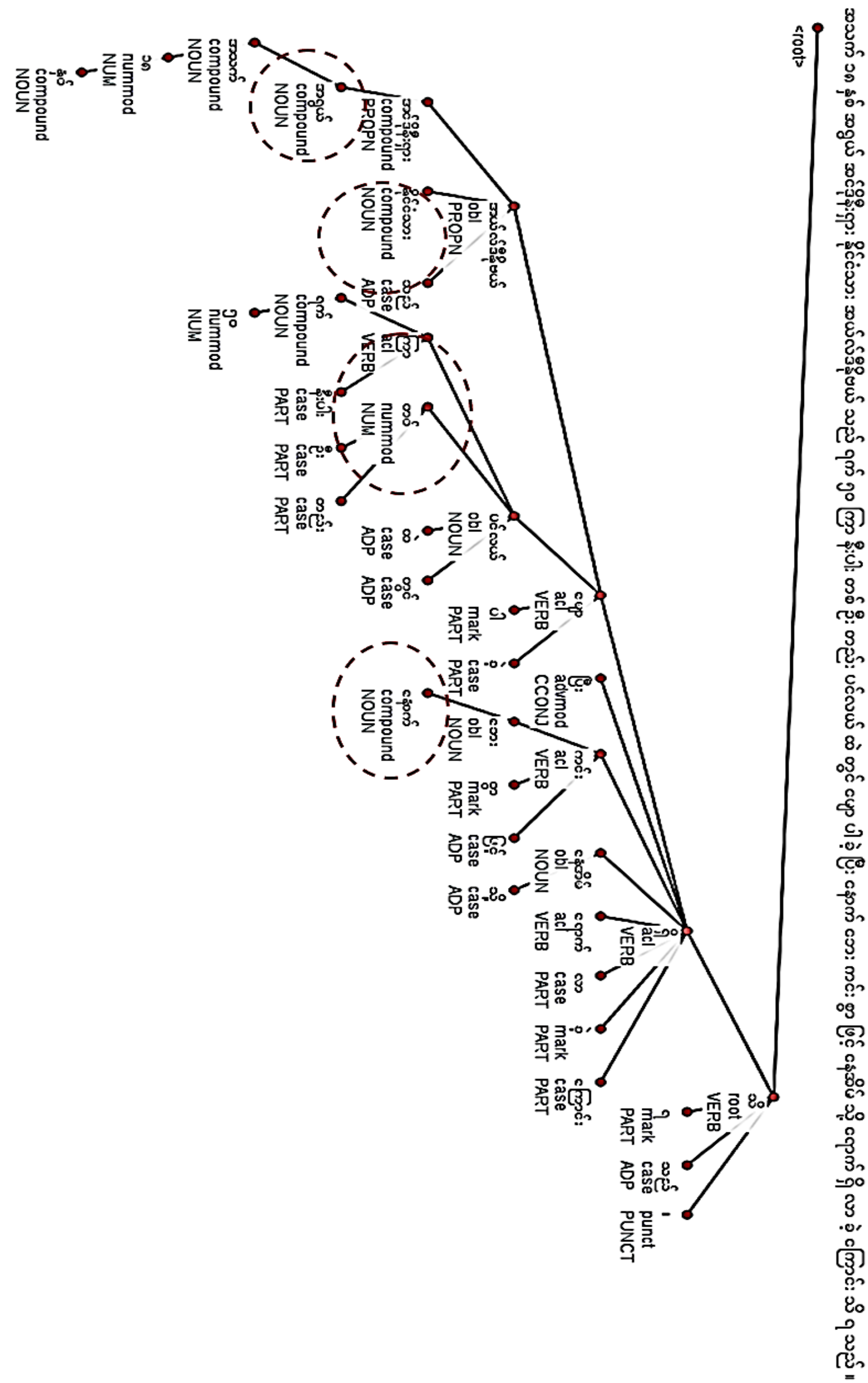
System model generated output



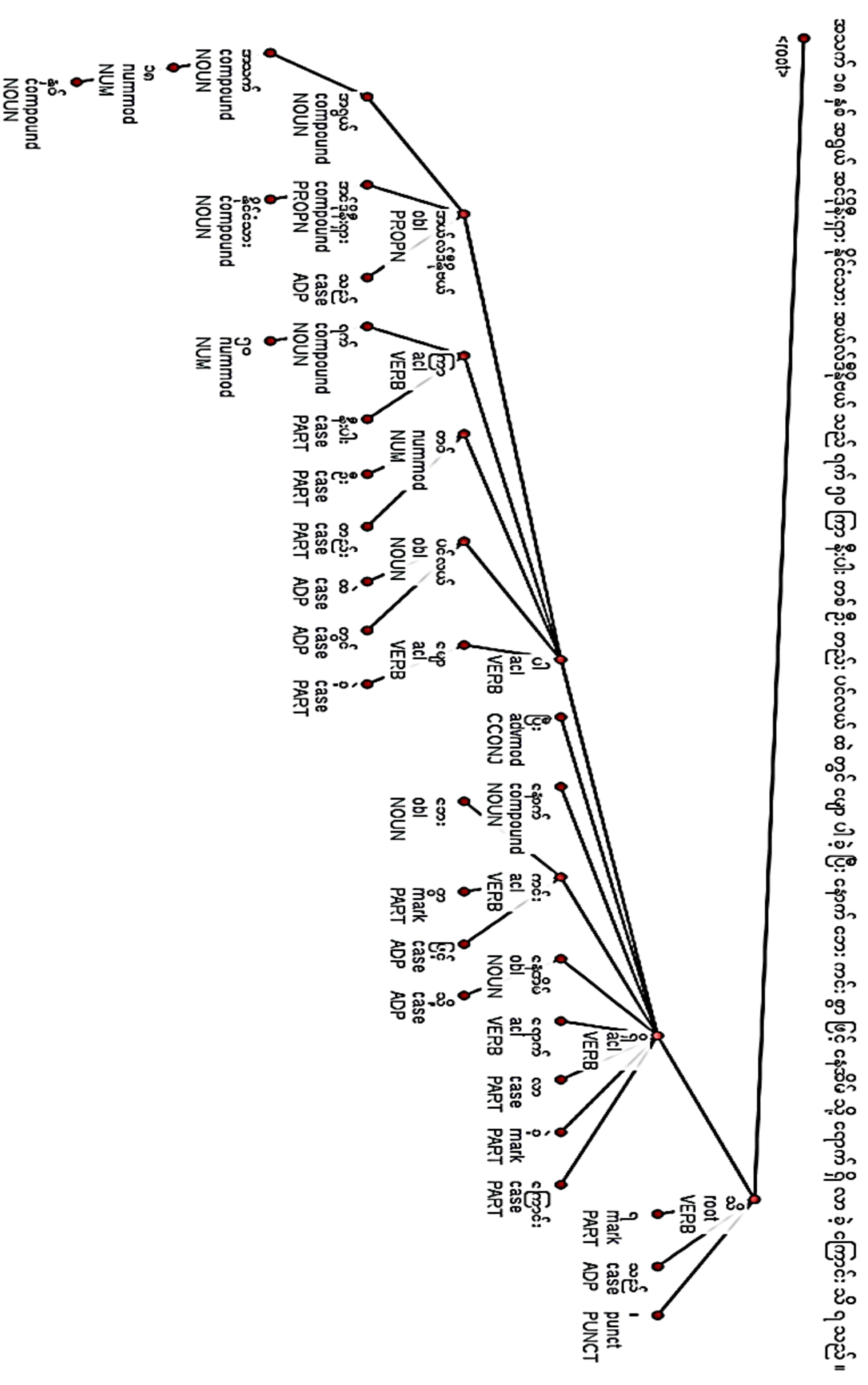
Reference Tree



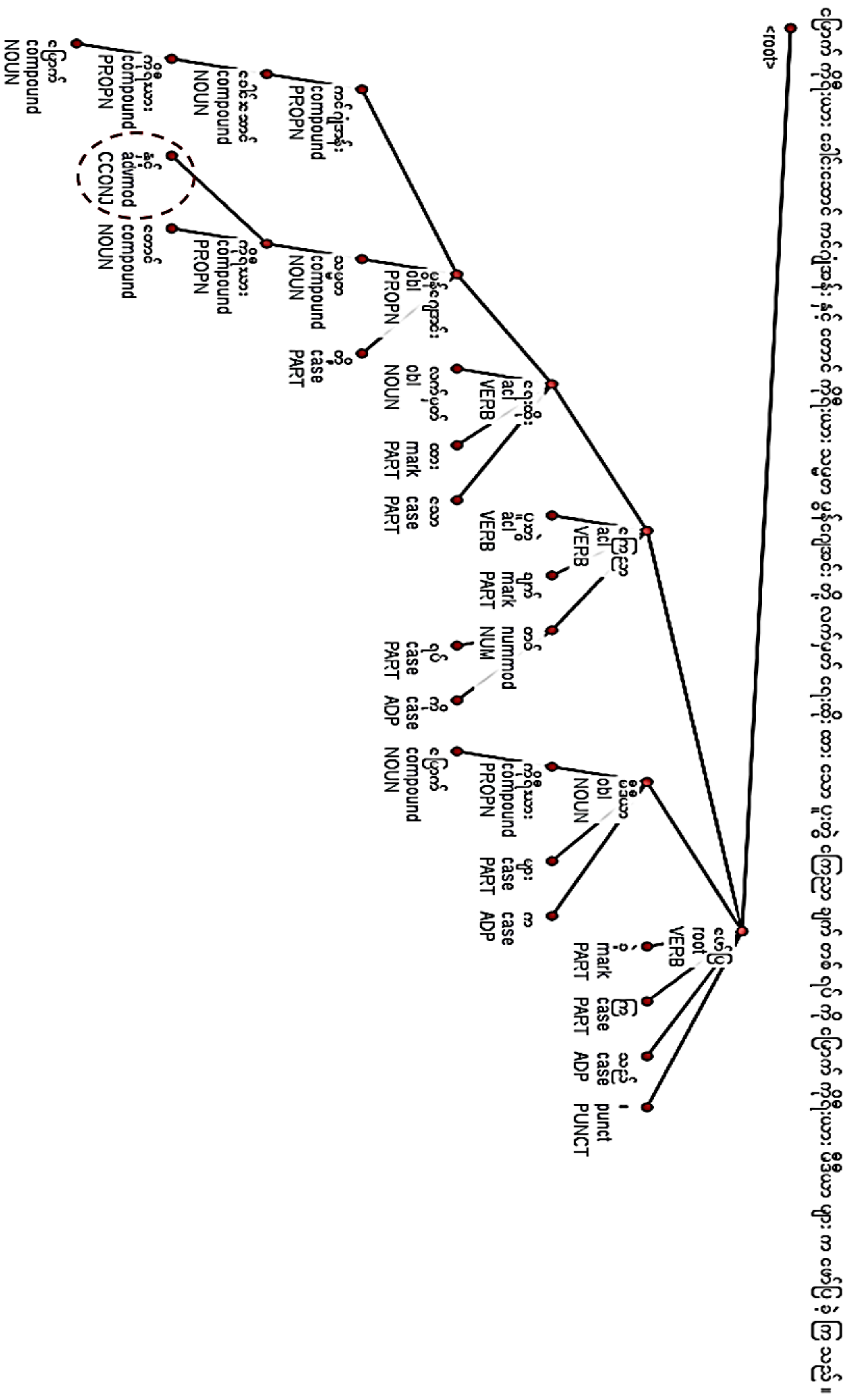
System model generated output



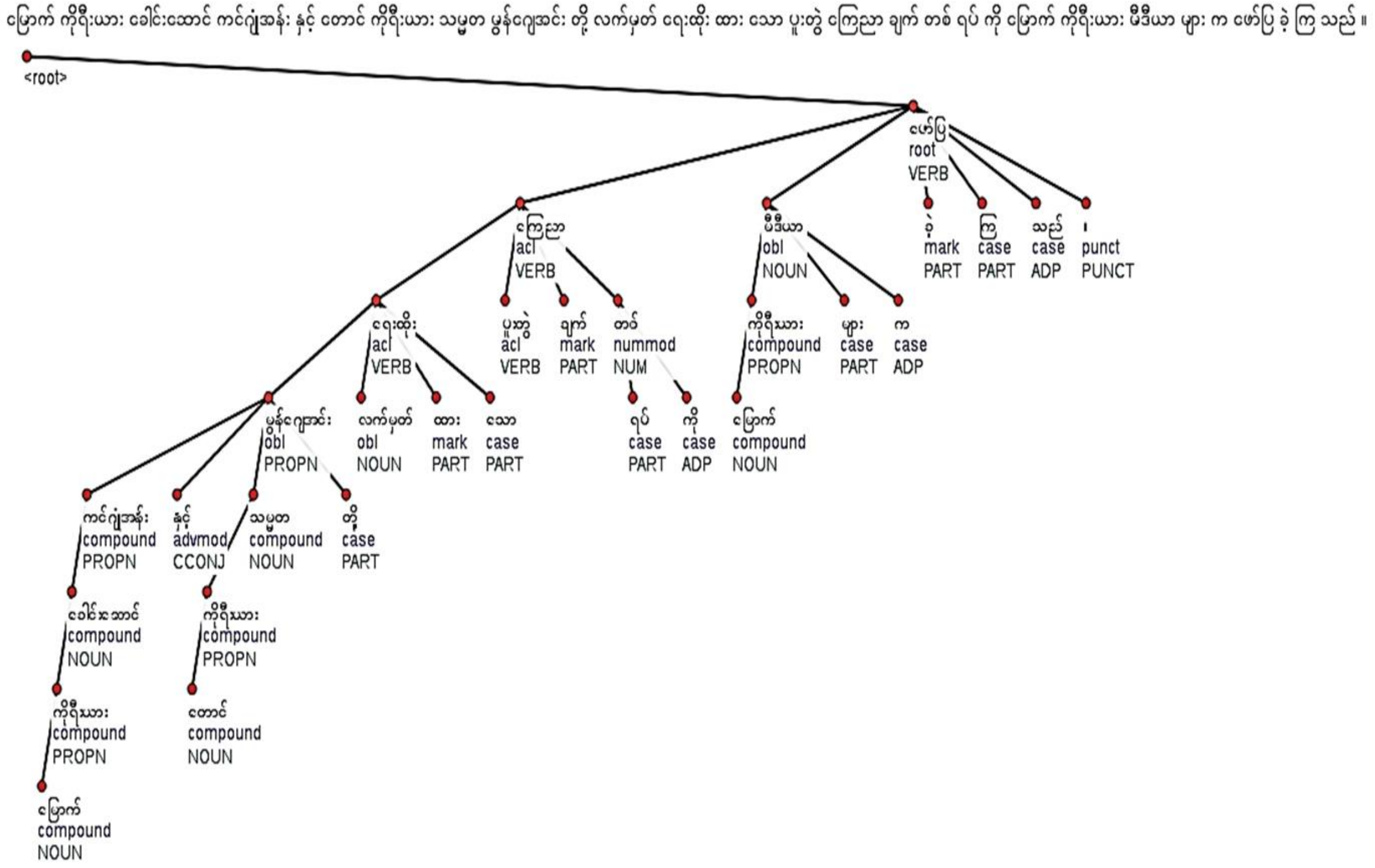
Reference Tree



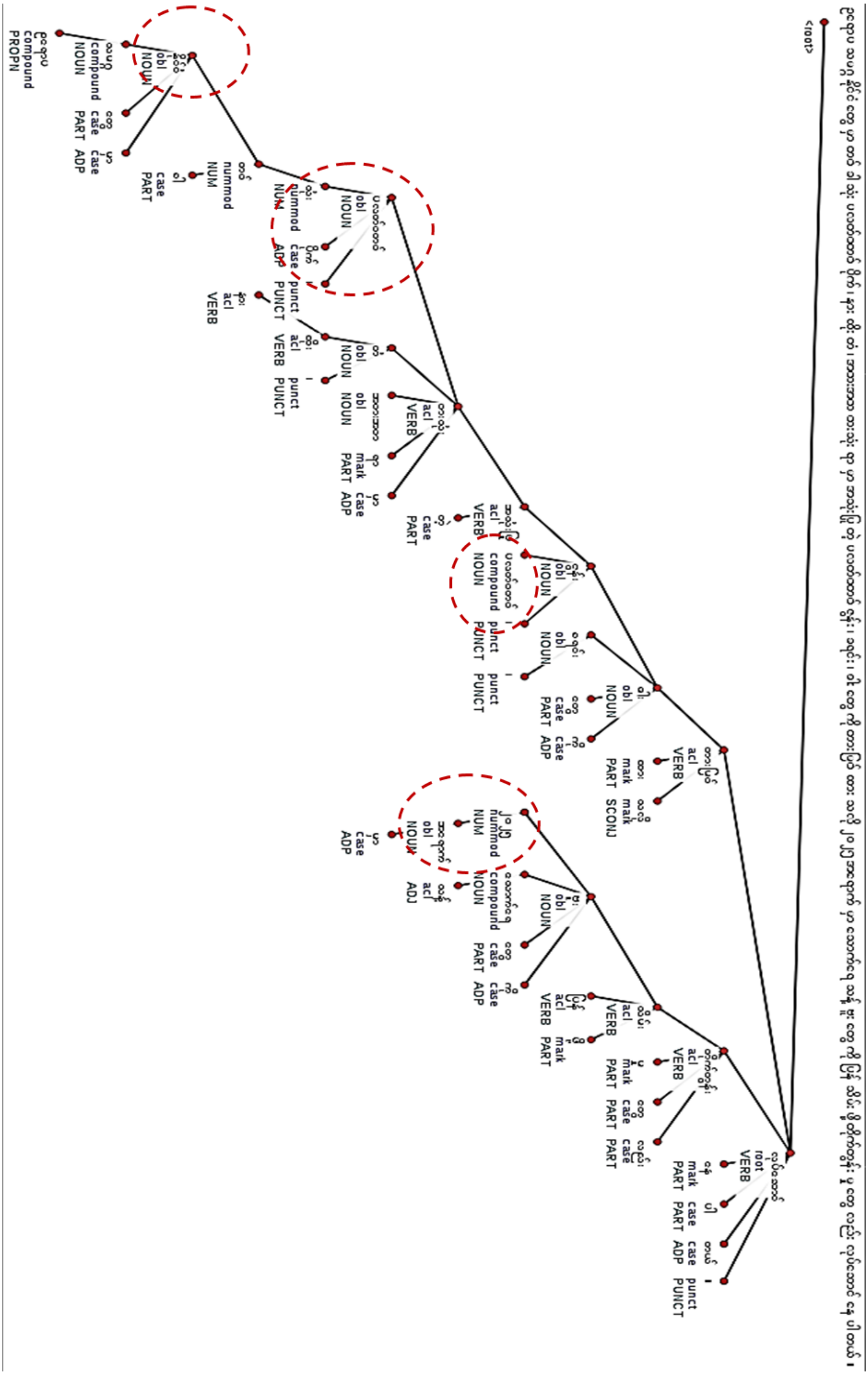
System model generated output



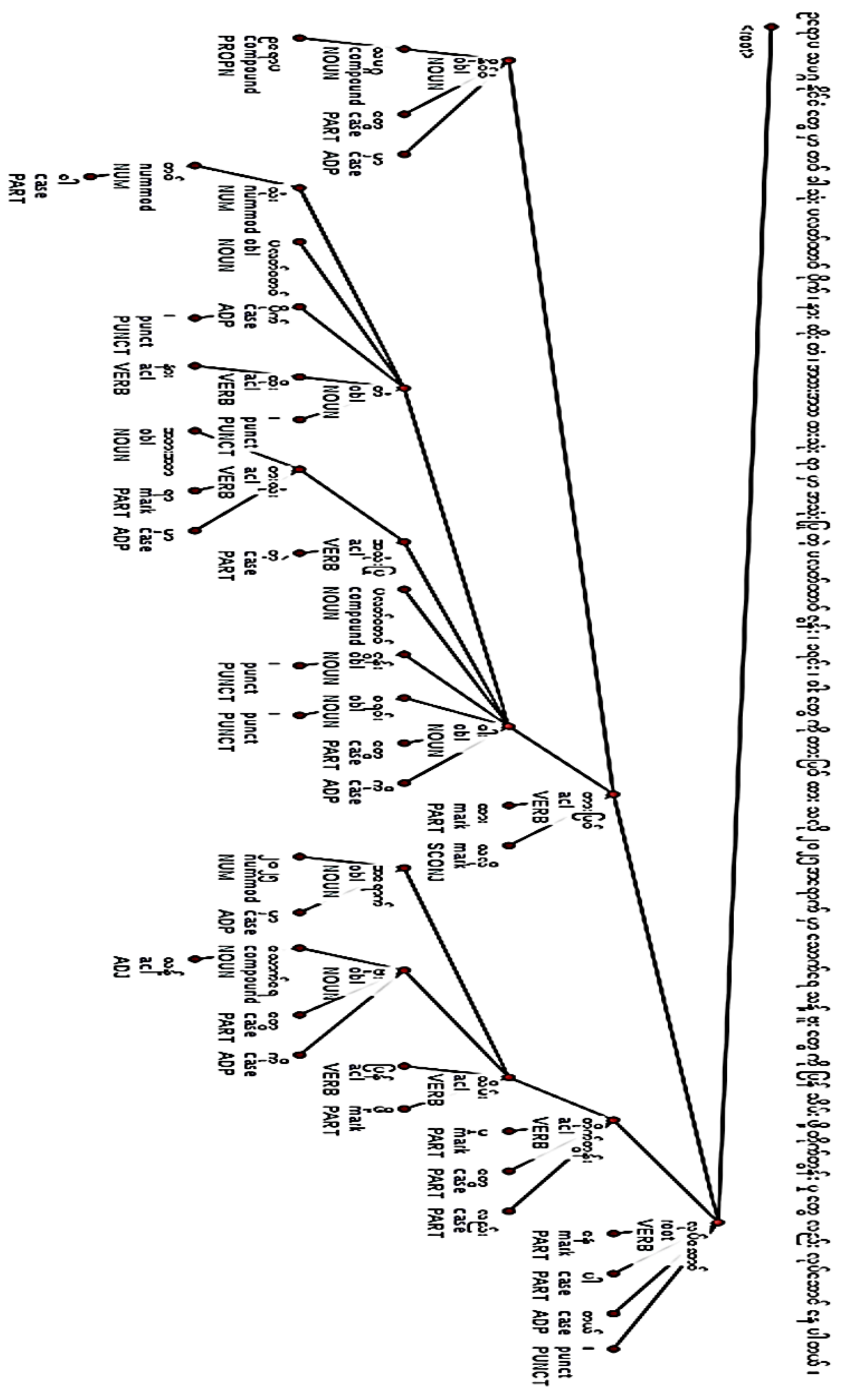
Reference Tree



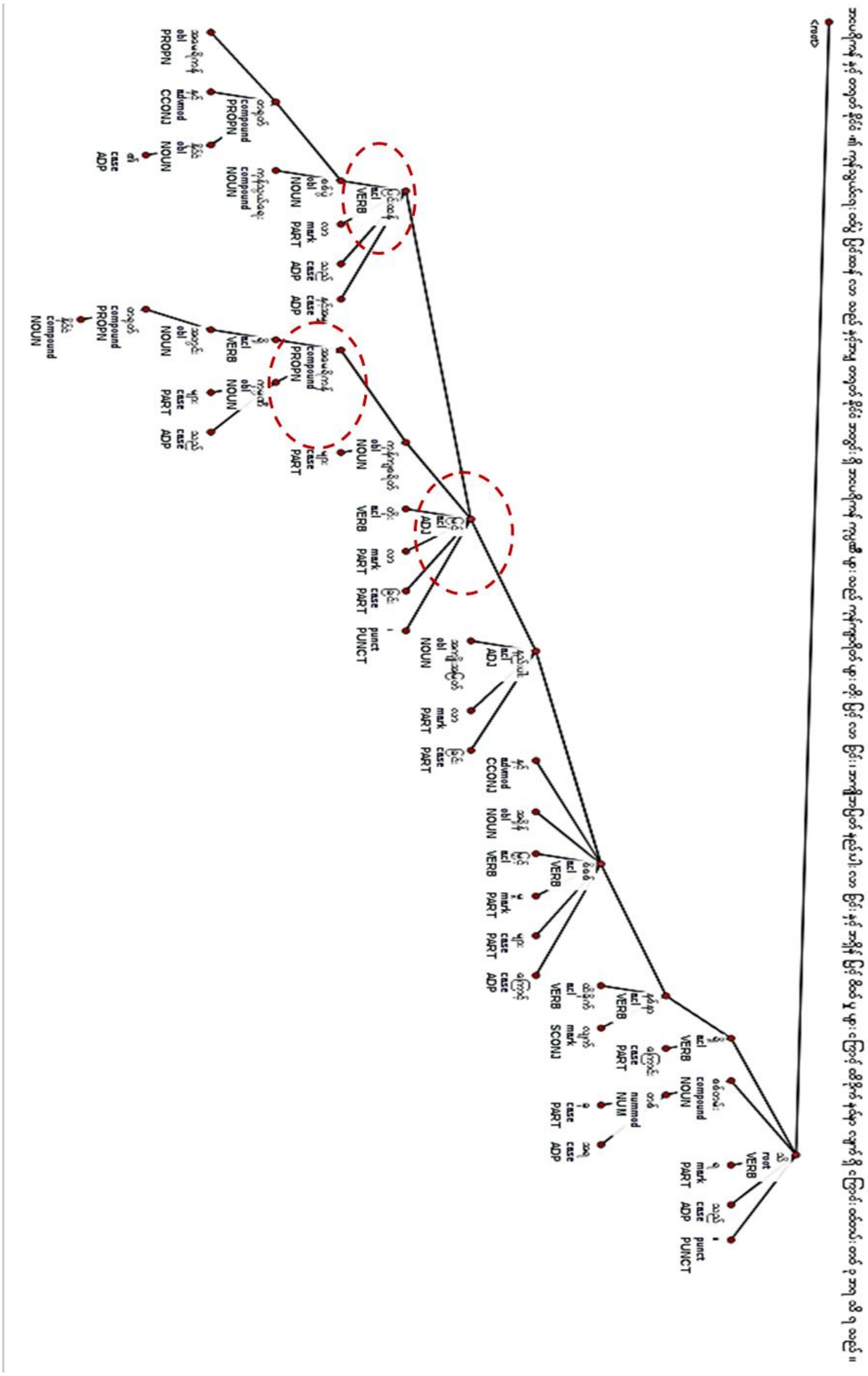
System model generated output



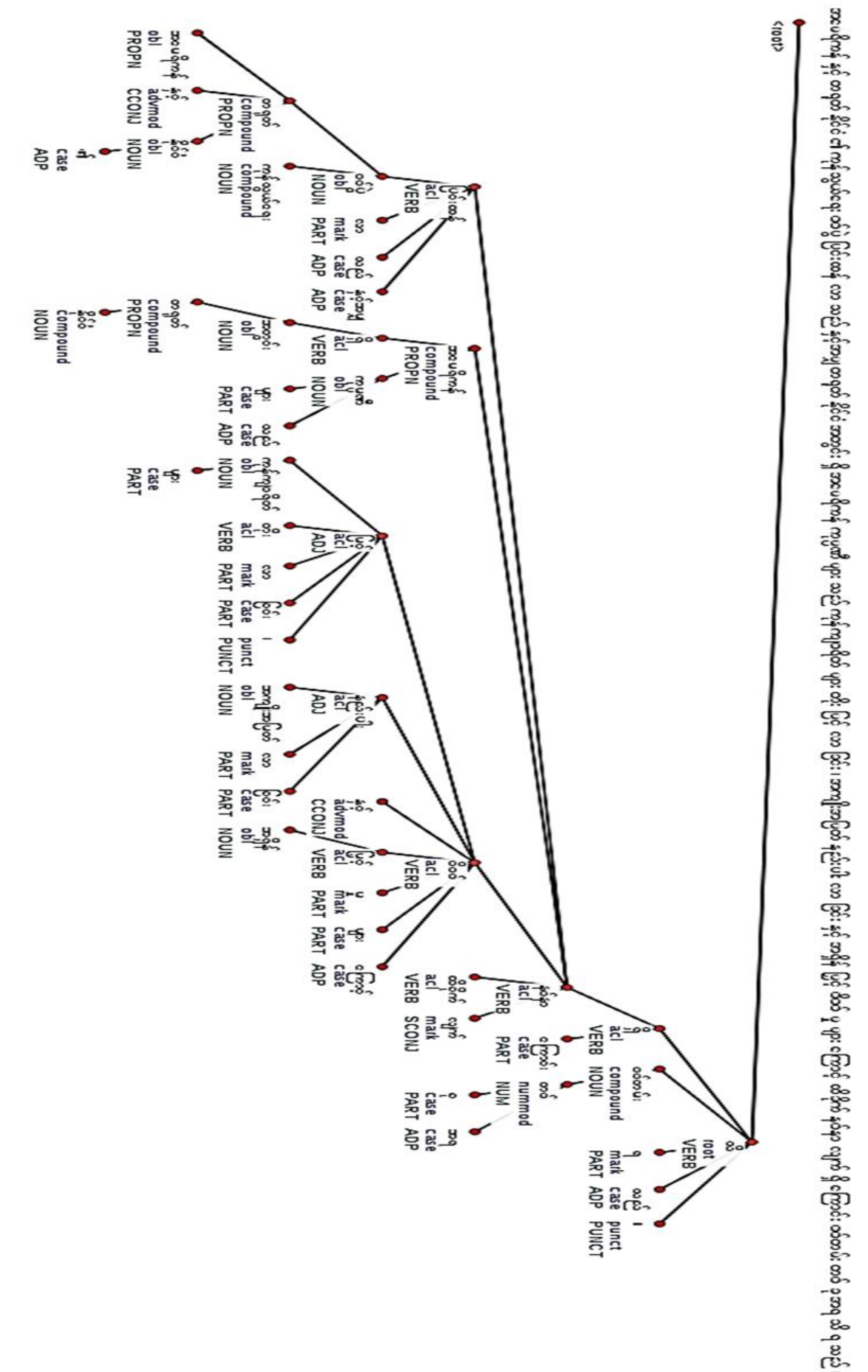
Reference Tree



System model generated output

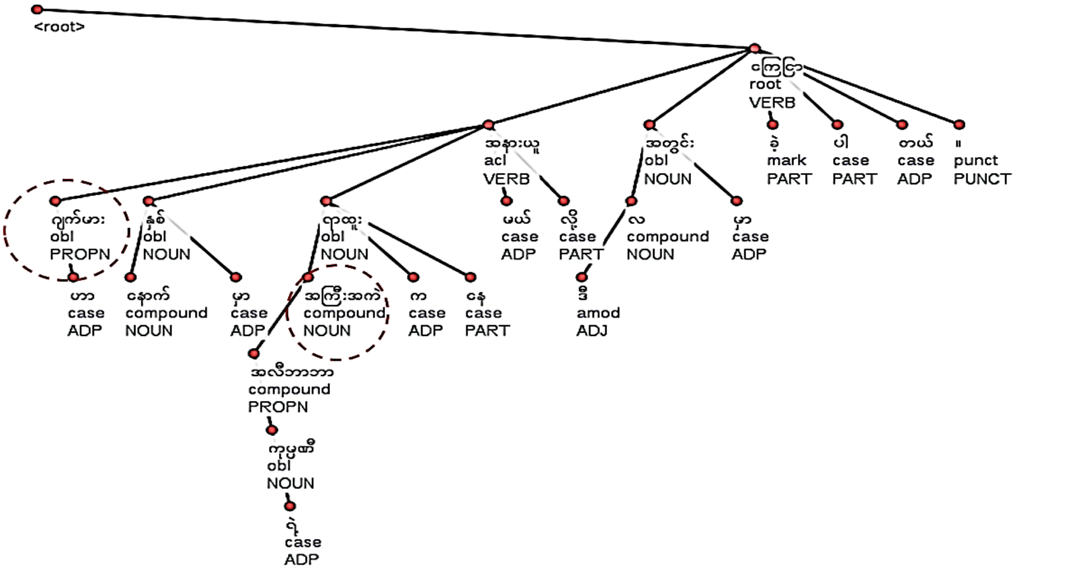


Reference Tree



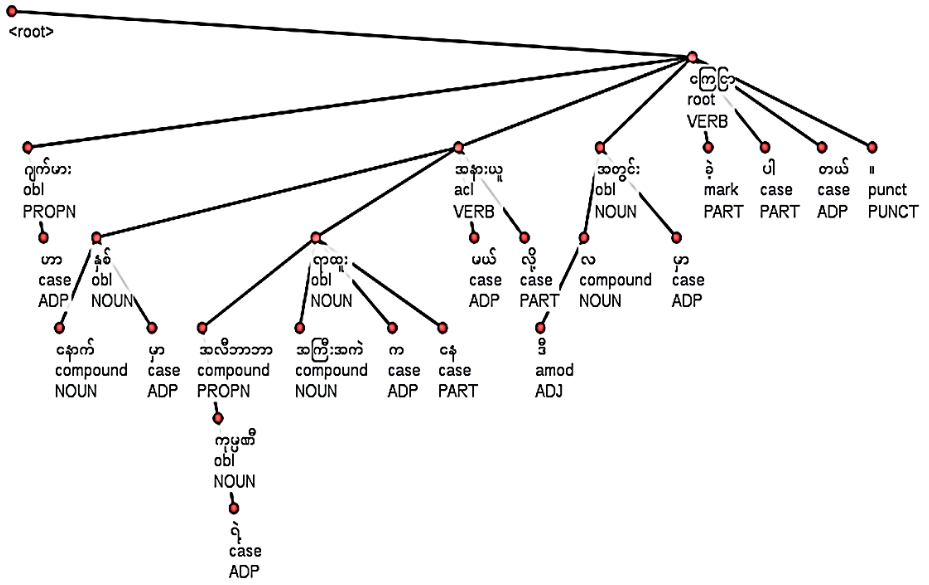
System model generated output

ဂျက်မား ဟာ နောက် နှစ် မှာ အလီဘာဘာ ကုပ္ပဏီ ရဲ့ အကြီးအကဲ ရာထူး က နေ အနားယူ ပယ် လို့ ဒီ လ အတွင်း မှာ ကြေငြာ ခဲ့ ပါ တယ် ။



Reference Tree

ဂျက်မား ဟာ နောက် နှစ် မှာ အလီဘာဘာ ကုပ္ပဏီ ရဲ့ အကြီးအကဲ ရာထူး က နေ အနားယူ ပယ် လို့ ဒီ လ အတွင်း မှာ ကြေငြာ ခဲ့ ပါ တယ် ။


<root>
ကြေငြာ root VERB
ဂျက်မား obl PROPN
အနားယူ acl VERB
အတွင်း obl NOUN
ခဲ့ mark PART
ပါ case PART
တယ် case ADP
။ punct PUNCT
ဟာ case ADP
နှစ် obl NOUN
ရာထူး obl NOUN
ပယ် case ADP
လို့ case PART
လ compound NOUN
မှာ case ADP
နောက် compound NOUN
မှာ case ADP
အလီဘာဘာ compound PROPN
အကြီးအကဲ compound NOUN
က case ADP
နေ case PART
ဒီ amod ADJ
ကုပ္ပဏီ obl NOUN
ရဲ့ case ADP