Context Free Grammar Based Top-Down Parsing of Myanmar Sentences

Win Win Thant, Tin Myat Htwe and Ni Lar Thein

Abstract— Parsing is important in Linguistics and Natural Language processing to understand the syntax and semantics of a natural language grammar. Writing the grammar production for Myanmar language is bit difficult because Myanmar is a relatively free word order, morphologically rich and agglutinative language and has a strong case marking system. This paper presents a context free grammar (CFG) based top-down parsing for Myanmar sentences which is the output of function tagging system. We use the function tags that are proposed in our previous function tagging method for producing the grammar rules. In this paper, we present successful parses of simple and complex Myanmar sentences.

Keywords—Context Free Grammar, Function Tags, Myanmar Sentences**,** Top-Down Parsing.

I. INTRODUCTION

PARSING natural language text is challenging because of the problems like ambiguity and inefficiency. It is considered to be an important intermediate stage for semantic analysis in natural language processing (NLP) application such as information retrieval (IR), information extraction (IE) and question answering (QA). It is the process of automatically building syntactic analysis of a sentence in terms of a given grammar and corpus. The study of natural language grammar dates back at least to 400 BC, when Panini described Sanskrit grammar, but the formal computational study of grammar can be said to start in the 1950s with work on context free grammar (CFG) [1].

The output of parsing is something logically equivalent to a tree, displaying dominance and precedence relation between constituents of a sentence. Much work has been done in different languages in different aspect of parsing, but most of these approaches cannot be applied to Myanmar language context. The main reason is most of the Myanmar language is highly inflectional, relatively free word order and agglutinative. Unlike fixed word order language such as English, in morphologically rich free word order languages the preferable linguistics rule set is too large. Apart from possessing all characteristics of a free word order language, Myanmar has some additional characteristics which make parsing a more difficult job. For example one or more than one suffixes are added with all relational constituents.

Our work is of the steps of Myanmar to English machine translation system. Since Myanmar grammar can be expressed in context-free form, the outcome of Myanmar sentence parsing can be represented as a parse tree.

W. W. Thant is with Natural Language Processing Laboratory, University of Computer Studies, Yangon, Myanmar (e-mail:winwinthant@gmail.com).

T. M. Htwe is with Natural Language Processing Laboratory, University of Computer Studies, Yangon, Myanmar (e-mail: tinmyathtwe@gmail.com).

N. L. Thein is with Natural Language Processing Laboratory, University of Computer Studies, Yangon, Myanmar (e-mail: nilarthein@gmail.com).

II. CHALLENGES OF PARSING

We always have many problems in natural language processing especially for syntax parsing.

A. Features of Myanmar Language

1) Myanmar as free word order language: Generally Myanmar sentence follows the subject, object, and verb pattern. However the interchange of subject, object is acceptable.For example: He buys computer.

သူ - ကွန်ပျူတာကို - ဝယ်သည်။
he - computer - buys
(or)
ကွန်ပျူတာကို - သူ - ဝယ်သည်။
computer - he - buys

In this example the first sentence is of the form subject, object and verb, whereas the second sentence is of the form object, subject and verb. This free word nature of Myanmar language is made possible by it being a morphologically rich language. In particular, free word order language like Myanmar local word groups help in determining syntactic function.

2) Myanmar as verb optional language: In Myanmar sentence, some phrases are optional. However if a verb component occurs, it normally occurs in the final position of the sentence. It is not necessary that all sentences have subject verb and object.

For example: He is Mg Hla.

သူက - မောင်လှ - ပါ။
He - Mg Hla - sentence's final particle

In this case the sentence's final particle ပါ (equivalent to "is" in English) is absent and is a meaningful sentence.

3) Myanmar as inflectional language: Case markers or postpositional markers (PPM) indicate thematic cases like subject, object etc. In Myanmar thematic cases are generally indicated by case suffixes attached to the noun itself. This

means that the case suffixes attached nouns can occur anywhere in the sentence. There are seventeen types of postpositional markers (PPM) in Myanmar language [2].For example: Ma Ma cooks the curry in the morning.

မမ - **သည်** - နံနက် - **တွင်** -
Ma Ma - **Subj PPM** - morning - **Time PPM** (in) -
ဟင်း - **ကို** - ချက်သည်။
curry - **Object PPM** - cooks

4) Myanmar as usage of particles language: The Myanmar language makes prominent usage of particles, which are untranslatable words that are suffixed or prefixed to words to indicate level of respect, grammatical tense, or mood [3]. For example: If Mg Mg wins the first prize, his parents will surprise.

မောင်မောင် - **များ** - ပထမ - ဆု - ရ -
Mg Mg - **particle** - first - prize - wins -
လျှင် - သူ့မိဘများ - က - အံ့သြ - လိမ့်မည်။
if - his parents - Subj PPM - surprise - will

B. The ambiguity of Myanmar syntax

In general we have many results for each Myanmar sentence. With ambiguous sentences we have more difficult for parsing phase. In fact, we have more ambiguities in spoken language, so it's very difficult to build a system that can understand meaning of whole sentence in every language forms. So, in this paper, we only make an examination of standard sentence.

C. The lack of necessary Myanmar linguistic data

In general, Myanmar to English text processing has not been deeply researched, there're not many results for Myanmar. We don't have much knowledge resources for Myanmar, so we have extremely difficulty in natural language processing especially syntax parsing.

1) Functional Annotated Corpus: In language teaching tools annotated corpus plays an important role. Annotation of corpora is needed in sentence level as well as in word level. In the sentence level, we have annotated corpus for parts of speech, chunk, and function tags relationship between the words in a sentence. In the word level, Lemma and morpheme annotation was done. Part of speech tagging and function tagging forms the basic step towards building a functional annotated corpus at sentence level.

We created functional annotated corpus in the previous function tagging system [4]. This work is being extended for additional 720 sentences to improve the performance further. These sentences are collected from Myanmar religious books, short stories and the new light of Myanmar newspaper and tagged its POS, chunk and function categories manually.

NC@PSubj[သူမ/pron.person]#PPC@SubjP[သည်/ppm.subj]# NC@PObj [လူနာ/n.person,များ/part.number]#PPC@ObjP[ကို/ppm.obj] #NC@PSim [ဆွေမျိုး/n.person,များ/part.number] #PPC@SimP[ကဲ့သို့/ppm.sim]#VC@ Active[ပြုစု/v.common] # SFC@Null[သည်/sf]။

Fig. 1: A sentence in the corpus

In our corpus, a sentence shown in Fig. 1 contains chunk types, function tags, Myannmar words and POS tags and categories. Example chunk types are Noun Chunk (NC), Postpostional Chunk (PPC) and Verb Chunk (VC). Example function tags are PSubj and SubjP for subject phrase, PObj and ObjP for object phrase and Active for verb phrase. Some POS tags and its categories are pron.person, ppm.subj, n.person, part.number, ppm.obj, ppm.sim and v.common.

III. PARSING MYANMAR SENTENCES

Parsing is the process of determining whether a string of tokens can be generated by a grammar. It breaks down words into functional units that can be converted into machine language. It also involves grouping the tokens of the source program into grammatical phrases that are used by the compiler to synthesize output. The grammatical phrases of the source program are represented by parse tree. A parse tree is a tree that represents the syntactic structure of a string according to some formal grammar.

A. Context Free Grammar

A context-free grammar is a formal system that describes a language by specifying how any legal text can be derived from a distinguished symbol called the sentence symbol. It consists of a set of productions, each of which states that a given symbol can be replaced by a given sequence of symbols [5]. A CFG has four components:

1. A set of tokens called terminals.
2. A set of variable called non terminals.
3. A set of production rules.
4. A designation of one of the non terminals as the start symbol.

B. Top Down Parsing

Top down parsing is one strategy that build parse from the start Symbol (S). Top Down parsing is goal oriented. The goal is towards to parse the sentence according to the grammar production. To build a parse, it repeats the following steps until the fringe of the parse tree matches the input string [6].

1. At the Start node S, Select a production with S on its left hand side and for each symbol on its right hand side, construct the appropriate child.
2. When a terminal is added to the fringe that doesn't match the input string, then backtrack.
3. Find the next node to be expanded.

If the parse tree did not match the input string then it means that input string is wrong.

C. Myanmar Grammar Rules

Designing a grammar for the entire Myanmar language is a daunting, difficult task. For the sake of simplicity, we will work with simple grammars that can generate only a subset of Myanmar by writing grammatical productions with CFG. We use the function tags that are proposed in [4]. There are 38 rules for function tags. The function tags are combined to get the phrases. There are 183 rules for grammatical relations of the phrases.

1) Rules for function tags

Subj → PSubj SubjP | PSubj CCC PSubj SubjP

Obj → PObj ObjP |Obj

Use → PUse UseP

Com → PCom CCC PCom ComP

Pla → PPla PlaP |Pla

2) Rules for phrases

Sent → Subj | Obj Subj Active |
Subj Obj Active | Subj Active |
Subj PcomplS Active
Sent CCS Sent | Obj Active

Obj → Sent CCP | Sent CCA Obj

Pla → Sent CCA Pla

IV. Parsing Simple Sentences

The function tagged sentence is an input into actual parsing process. For the simple sentence, adjectives and adverbs are attached with their corresponding noun phrase and verb phrase in the sentence [7]. Then noun phrase and verb phrase are attached with the root of the tree. The simple sentence can be extended with many noun phrases and one verb phrase. Noun phrase may be subject phrase, object phrase, place phrase and so on. For example:

သူသည် A နီရောင်စာA ုပ်ကို ဆရာA ၁း ကျောင်းတွင် ရိုသေစွာပေးသည်။

(He gives the red book to the teacher in the school respectfully).

The following sentence is the function tagged sentence of the above sentence. It is the output of previous function tagging system.

PSubj[သူ]#SubjP[သည်]#PObj[A နီရောင်စာA ုပ်]#ObjP[ကို]#PIobj[ဆရာ့]
#IobjP[A ၁း]#PPla[ကျောင်း]#PlaP[တွင်]#Active [ရိုသေစွာပေးသည်]

The function tags are PSubj, SubjP for subject (Subj) phrase, PObj and ObjP for object (Obj) phrase, PIobj and IobjP for indirect object (Iobj) phrase, PPla and PlaP tag for place (Pla) phrase and Active for verb phrase.

After parsing with CFG grammar, the output parse tree is shown in Fig. 3. In this figure, the Myanmar sentence (Sent) is made up of Subj, Obj, Iobj, Pla and Active phrases. In English, Subj (He), Obj (red book), Iobj (to the teacher), Pla (in the school) and Active (gives respectfully) are included in the sentence.

Sent→**Subj Obj Iobj Pla Active**
→**PSubj SubjP** Obj Iobj Pla Active
→PSubj SubjP **PObj ObjP** Iobj Pla Active
→PSubjSubjP PObj ObjP **PIobj IobjP** PlaActive
→PSubj SubjP PObj ObjP PIobj IobjP **PPla PlaP** Active

Fig. 2. Top down derivation for the simple sentence

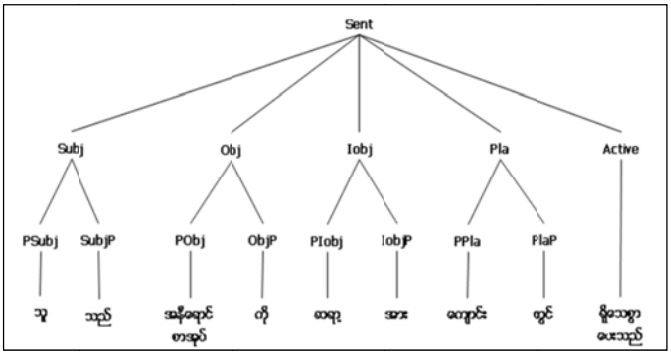

Fig. 3. Parse tree for the simple sentence

V. Parsing Complex Sentences

Complex sentences with more than one verb with conjunctions can be parsed by the following steps.

1. Conversion of complex sentence into simple sentence by eliminating the conjunctions.
2. Parse these simple sentences as sub sentences of the main sentence separately and join these parsed sub sentences with conjunctions to form the syntactic constituents of the main sentence.

There are three types of complex sentences according to the conjunctions (particles, adjective and adverb).

A. Complex Sentences joined with particles (CCP)

1) One sentence is the Object of other sentence: ကျွန်မ ဘီစကွတ်မုန့် စားသည် **ကို** A မေ မြင်သည်။ (Mother sees **that** I eat biscuit.)

There are two simple sentences: (1) A မေ မြင်သည် (Mother sees) and (2) ကျွန်မ ဘီစကွတ်မုန့် စားသည် (I eat biscuit). In our language, these two simple sentences are joined with ကို (particle) and became a complex sentence.

PSubj[ကျွန်မ]#SubjP[ဘီစကွတ်မုန့်]#Active[စားသည်]#CCP[ကို]#Subj
[A မေ]#Active[မြင်သည်] (1)

If the above function tagged sentence is divided into two sentences by eliminating the conjunction (ကို/CCP), there exist two simple sentences. According to our language feature, the whole first sentence is the Object of the second one (see Fig. 5).

After top down parsing with CFG grammar, the output parse tree is shown in Fig. 5. In this figure, the Myanmar sentence (Sent) is made up of Obj, Subj and Active phrases. Obj is made up of Sent and CCP. Sent is made up of Sub, Obj and Active.

Sent →**Obj Subj Active**
→**Sent CCP** Subj Active
→ **Subj Obj Active** CCP Subj Active

Fig. 4. Top down derivation for the sentence (1)

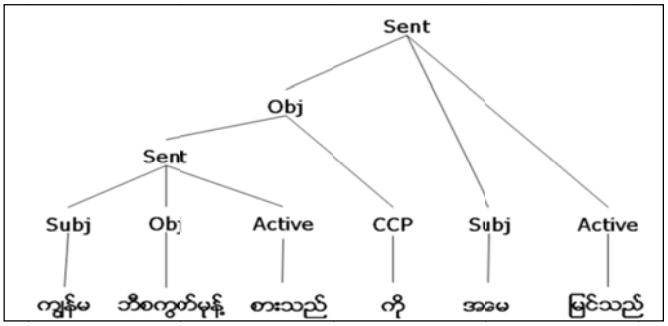

Fig. 5. Parse tree for the sentence (1)

B. Complex Sentences joined with adjective (CCA)

1) Two sentences with one middle sentence: ကျွန်မ သည် A မေ ပေး **သော** မုန့် ကို စားသည်။ (I eat the snack **that** is given by mother.)

There are two simple sentences: (1) ကျွန်မ မုန့် စားသည် (I eat the snack) and (2) A မေ မုန့်ကို ကျွန်မA ား ပေးသည် (Mother gives me the snack). In our language, these two simple sentences are joined with သော (adjective) and became a complex sentence.

PSubj[ကျွန်မ]#SubjP[သည်]#Subj[A မေ]#Active[ပေး]#CCA[သော]# PObj[မုန့်]#ObjP[ကို]#Active[စားသည်] (2)

If the above function tagged complex sentence is divided into two sentences by eliminating the conjunction (သော/CCA), there exist two simple sentences. The first sentence has two consecutive Subj tags and the second one starts with Obj tag. So, one sentence is in the middle of the other sentence as an Object (see Fig. 7).

Sent →**Subj Obj Active**
→**PSubj SubjP** Obj Active
→PSubj SubjP **Sent CCA Obj** Active
→ Subj SubjP **Subj Active** CCA Obj Active
→PSubj SubjP Subj Active CCA **PObj ObjP** Active

Fig. 6. Top down derivation for the sentence (2)

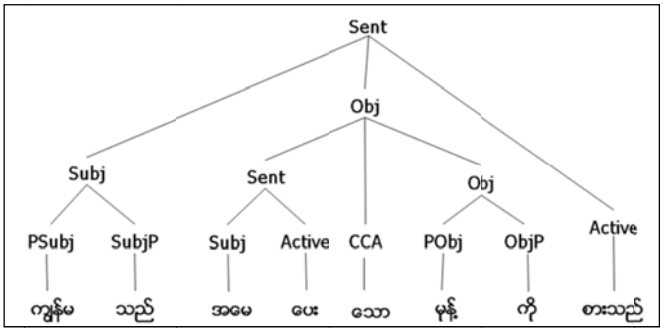

Fig. 7. Parse tree for the sentence (2)

2) One sentence is the modifier of the Subject of other sentence: ကျွန်မ စား သော မုန့် သည် ဘီစကွတ်မုန့် ဖြစ်သည်။ (The snack that I eat is biscuit.)

There are two simple sentences: (1) ကျွန်မ မုန့် စားသည် (I eat the snack) and (2) မုန့် သည် ဘီစကွတ်မုန့် ဖြစ်သည် (The snack is biscuit). In our language, these two simple sentences are joined with သော (adjective) and became a complex sentence.

Subj[ကျွန်မ]#Active[စား]#CCA[သော]#PSubj[မုန့်]#SubjP[သည်]# PcomplS[ဘီစကွတ်မုန့်]#Active[ဖြစ်သည်] (3)

If this function tagged sentence is divided into two sentences by eliminating the conjunction (သော/CCA), there exist two simple sentences. The first sentence is Subj Verb pattern, the conjunction is CCA and the second one starts with Subj tag. So, the first sentence is the modifier of the Subject of the second sentence (see Fig. 9).

Sent →**Subj PcomplS Active**
→**Sent CCA Subj** PcomplS Active
→**Subj Active** CCA Subj PcomplS Active
→Subj Active CCA **PSubj SubjP** PcomplS Active

Fig. 8. Top down derivation for the sentence (3)

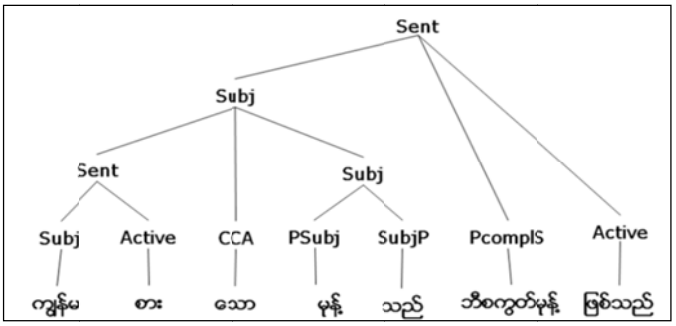

Fig. 9. Parse tree for the sentence (3)

C. Complex Sentences joined with adverb (CCS)

1) Two sentences with only one subject: ကျွန်မ သည် စာဖတ် **ပြီးနောက်** မုန့် ကို စားသည်။ (I read **and** eat the snack.)

There are two simple sentences: (1) ကျွန်မ စာဖတ် သည် (I read) and (2) ကျွန်မ မုန့် ကို စားသည် (I eat the snack). In our language, these two simple sentences are joined with ပြီးနောက် (adverb) and became a complex sentence.

PSubj[ကျွန်မ]#SubjP[သည်]#Active[စာဖတ်]#CCS[ပြီးနောက်]#PObj[မုန့်]# ObjP[ကို]#Active[စားသည်] (4)

If this function tagged sentence is divided into two sentences by eliminating the conjunction (ပြီးနောက်/CCS), there exist two simple sentences. The first sentence is Subj Verb pattern, the conjunction is CCS tag and the second one will not have the subject. So, the subject of the first sentence can be considered as its subject (see Fig. 11).

Sent → **Sent CCS Sent**
→ **Subj Active** CCS Sent
→ **PSubj SubjP** Active CCS Sent
→ PSubj SubjP Active CCS **Obj Active**
→ PSubj SubjP Active CCS **PObj ObjP** Active

Fig. 10. Top down derivation for the sentence (4)

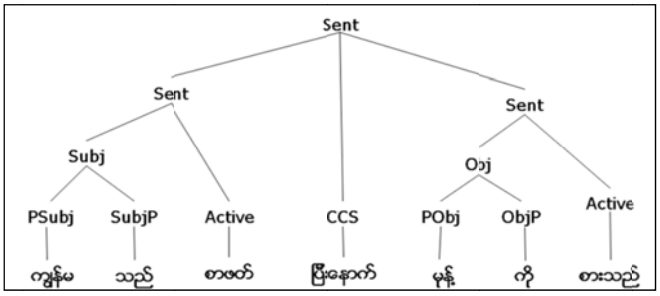

Fig. 11. Parse tree for the sentence (4)

VI. Evaluation

There are nearly 3000 training sentences and 530 testing sentences. The sentences consist of 5 to 50 words. We divided sentences into simple and complex sentences. The simple sentences are declarative, negative and interrogative. Three types of complex sentences are joined with particles, adjectives and adverbs respectfully. We named sentence type from 1 to 6. Sentence type 1 is declarative simple sentence. Type 2 is simple negative sentence. Type 3 is interrogative simple sentence. Type 4 is complex sentences joined with particles. Type 5 is complex sentences joined with adjectives. Type 6 is complex sentences joined with adverb. The sentences are tested and the output parse trees are manually checked. The accuracy of parse tree is calculated by using the following equation and shown in Table 1 and Fig. 12.

$$\text{Accuracy} = \frac{NumberOfCorrectParseTrees}{NumberOfTestSentences} \times 100\%$$

TABLE I
ACCURACY FOR DIFFERENT SENTENCE TYPES

Type	Trained sentences	Test sentences	Correct parse trees	Sentences Accuracy (%)
1	830	120	113	95.8
2	250	60	53	88.3
3	540	90	83	92.2
4	350	70	59	84.3
5	420	80	72	90
6	480	110	101	91.8
Total	2870	530	481	90.6

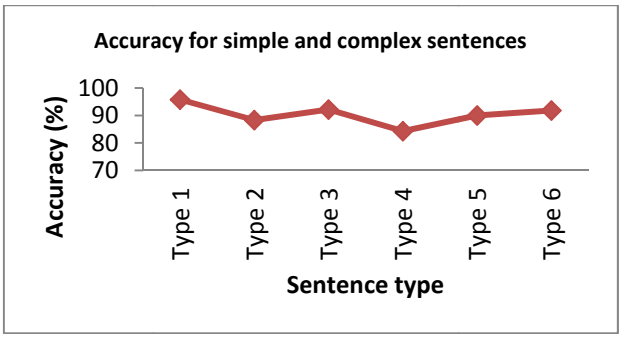

Fig. 12. Accuracy for different sentence types

VII. Conclusion and Future Work

The context free grammar for Myanmar sentence parsing has been introduced. We described CFG because it is easier to maintain, can add new language features and automatically construct efficient parser. In the process of producing grammar, we have used the function tags and developed 38 rules for function tags and 183 rules for phrases. We chose top-down parsing because it does well if there is useful grammar-driven control: search is directed by the grammar. This work is being extended for additional 720 sentences to improve the performance further. We parsed for simple sentences and complex sentences joined with two simple sentences. In future this parsing system will be developed with more than 3000 sentences to have the functional relationship among the words in the sentences which will lead to the best performance in the application of long term relationship and free word order. And we will be developed for complex sentences joined with many simple sentences.

References

[1] E. Charniak, "Statistical Parsing with a Context-Free-Grammar and Word Statistics," *In Proceedings of the 14th National Conference on Artificial Intelligence and 9th Innovative Applications of Artificial Intelligence Conference (AAAI-97/IAAI-97)*, pages 598-603, Menlo Park, July 27-31 1997. AAAI Press.

[2] Myanmar Thudda, vol. 1 to 5 in Bur-Myan, Text-book Committee, Basic Edu., Min. of Edu., Myanmar, ca. 1986.

[3] S. P. Soe, "Aspects of Myanmar Language", Myanmar Department, University of Foreign Language, 2010.

[4] W.W. Thant, T. M. Htwe, and N. L. Thein, "Function Tagging for Myanmar Language", International Journal of Computer Applications (IJCA), Volume 26, Number 2, July 2011.

[5] M.-J. Nederhof and G. Satta, "Parsing Non-recursive Context-free Grammars" ,In 40th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, pages 112-119, Philadelphia, Pennsylvania, USA, July 2002.

[6] Bala sundara Raman L, Ishwar S, Sanjeeth Kumar Ravindranath, "Context Free Grammar for Natural Language Constructs - An Implementation for Venpa class of Tamil Poetry" , Tamil Internet 2003, Chennai, India

[7] K. Lay, "Construction of Myanmar Thudda", Ph.D. Dissertation, Myanmar Department, University of Education, 2003.

[8] K. Min and William H. Wilson, "Are Efficient Natural Language Parsers Robust?", Proceedings of the Eighth Australian Joint Conference on Artifical Intelligence (AI '95), Canberra 13-17 November 1995, 283-290. Edited Xin Yao. World Scientific, Singapore, 1995. ISBN 981-02-2484-2