A First Step Towards Parsing of Assamese Text

Navanath Saharia
Department of CSE
Tezpur University
Assam, India 784028
nava.nath@yahoo.in

Utpal Sharma
Department of CSE
Tezpur University
Assam, India 784028
utpal@tezu.ernet.in

Jugal Kalita
Department of CS
University of Colorado
Colorado Springs, USA 80918
kalita@eas.uccs.edu

***Abstract*—Assamese is a relatively free word order, morphologically rich and agglutinative language and has a strong case marking system stronger than other Indic languages such as Hindi and Bengali. Parsing a free word order language is still an open problem, though many different approaches have been proposed for this. This paper presents an introduction to the practical analysis of Assamese sentences from a computational perspective rather than from linguistics perspective. We discuss some salient features of Assamese syntax and the issues that simple syntactic frameworks cannot tackle.**

***Keywords*-Assamese, Indic, Parsing, Free word order.**

I. Introduction

Like some other Indo-Iranian languages (a branch of Indo-European language group) such as Hindi, Bengali (from Indic group), Tamil (from Dardic group), Assamese is a morphologically rich, free word order language. Apart from possessing all characteristics of a free word order language, Assamese 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. Research on parsing model for Assamese language is purely a new field. Our literature survey reveals that there is no annotated work on Assamese till now.

In the next section we will present a brief overview of different parsing techniques. In section III we discuss related works. Section IV contains a brief relevant linguistic background of Assamese language. In section V we discuss our approach we want to report in this paper. Section VI conclude this paper.

II. Overview of parsing

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). Parsing is a problem in many natural languages processing tasks such as machine translation, information extraction, question answering etc. It is the process of automatically building syntactic analysis of a sentence in terms of a given grammar and lexicon; and syntax is the name given to the study of the form, positioning, and grouping of the elements that go to make up sentences. The result may be used as input to a process of semantic interpretation. The output of parsing is something logically equivalent to a tree, displaying dominance and precedence relation between constituents of a sentence. Now-a-days there are several dimensions to characterize the behaviour of parsing technique, for example- depending on search strategy (such as Top-down, bottom-up parsing), statistical model used (such as Maximum Entropy model), Grammar formalism used (such as Paninian framework) etc. Among them most successful linguistically motivated formalisms are- Combinatory Categorial Grammar (CCG), Dependency Grammar(DG)[1], Lexical Functional Grammar (LFG) [2], Tree-Adjoining Grammar (TAG) [3], Head-Driven Phrase Structure Grammar (HPSG) [4], Paninian Grammar (PG) [5] and Maximum Entropy model (EM) [6].

III. Existing work

Reference [7], reported (Table I) word order variability that some language allow.

TABLE I
Word order variation table.

Almost no variation	English, Chinese, French
Some variation	Japanese, German, Finnish
Extensive variation	Russian, Korean, Latin
Maximum variation	Warlpiri

Our literature survey reveals that a majority of the parsing techniques are developed solely for the English language and might not work for other languages.Much work has been done in different languages in different aspect of parsing, but most of these approaches can not be applied to Indian language context. The main reason is most of the Indian languages are 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, which may not be handled using the approaches like PSG, LFG[2] etc. Among the reported formalisms, only CCG, PG and DG have literal evidence to apply on free word order languages.

An approach for Indian language parsing is Paninian framework which was developed in IIT, Kanpur. First it was designed only for free word order languages basically Hindi, afterward it was extended to other free word order language

The Department of Computer science & Engineering, Tezpur University is funded by university grant commission (UGC)'s departmental research support (DRS) Phase-I under special assistance programme (SAP).

such as Bangla, Tamil etc., but no attempt was made to build a parser for Assamese.

Among the more recent works [8], [9], [10] has focus on dependency parsing. Dependency grammar is an asymmetrical relation between a head and a dependent. Dependency grammar is a set of rules that describes the dependencies. Every word (dependent) depends on another word (head), except one word which is the root of the sentence.Thus a dependency structure is a collection of dependencies for a sentence and dependency parsing depends critically on predicting head-modifier relationship.

A classifier based dependency parser was proposed by Sagae and Lavie [11], that produces a constituent tree in linear time. The parser uses a basic bottom-up shift-reduce stack based parsing algorithm like Nivre and Scholz[12] but employs a classifier to determine parser actions instead of a grammar. Like other deterministic parsers (unlike other statistical parser), this parser considers the problem of syntactic analysis separately from part-of-speech (POS) tagging. Because the parser greedily builds trees bottom-up in a single pass, considering only one path at any point in the analysis, the task of assigning POS tags to word is done before other syntactic analysis. This classifier based dependency parser shares similarities with the dependency parser of Yamada and Matsumoto [13] that it uses a classifier to guide the parsing process in deterministic fashion, while Yamada and Matsumoto uses a quadratic run time algorithm with multiple passes over the input string.

A language-wise survey (Table II) shows that Nivre's parser was implemented in a variety of languages, like relatively free word order language (Turkish), inflectionally rich language (Hindi), fixed word order language (English), and relatively case-less and less inflectional language (Swedish), whereas Paninian grammar framework was implemented only for Indian language context and CCG approach was implemented for Dutch, Turkish and English Language. Other mostly implemented parsers are Collin's and Mc-Donald's parser.

TABLE II
LANGUAGE-WISE SURVEY OF IMPLEMENTED PARSER.

Nivre's Parser	English[12] Czech[14] Swedish[15] Chinese[16] Bulgarian[17] Turkish[18] Hindi[8]
Collin's Parser	English[19] Czech[20] Spanish[21] Chinese[22] German[23]
Mc Donald's Parser	English[24] Czech[24] Danish[25]
CCG Framework	English[26] Dutch[27] Turkish[28]

IV. ASSAMESE AS A FREE WORD ORDER LANGUAGE

For most languages that have a major class of nouns, it is possible to define a basic word order in terms of subject(S) verb(V) and object(O). There are six theoretically possible basic word orders: SVO, SOV, VSO, VOS, OVS, and OSV. Of these six, however, only the first three normally occur as dominant orders. If constituents of a sentence can occur in any order without affecting the gross meaning of the sentences (the emphasis may be affected) then that type of language is known as free word order language. Warlpiri, Russian, Tamil are the example of free word order language.

Typical Assamese sentences can be divided into two parts: Subject(S) and Predicate(P). Predicate may again be divided into following constituents- object(O), verb(V), extension(Ext) and connectives(Cv). A minimum sentence may consist of any one of S, O, V, Ex or even in a connected discourse. Table III shows some single constituent sentences of Assamese. Table IV shows, some two-constituent sentences that may also occur in any order.

TABLE III
SINGLE CONSTITUENT SENTENCES. (TF: TRANSLITERATED ASSAMESE FORM, ET: APPROXIMATE ENGLISH TRANSLATION)

N —	নমস্কাৰ।	TF: *namoskAr.*	
PN—	মই।	TF: *mai*	ET: I.
V —	আহা।	TF: *ahA*	ET: come.
PP—	আৰু।	TF: *aAru*	ET: and.

TABLE IV
TWO CONSTITUENT SENTENCES.

PN+V	মই আহিছো। TF: *mai aAhiso* EF: I have come.	V+PN	আহিছো মই। TF: *aAhiso mai*
N+V	কিতাপখন পঢ়িলো। TF: *kitApkhan parhilo* EF: (I) have read the book.	V+N	পঢ়িলো কিতাপখন। TF: *parhilo kitApkhan*
Adj+V	ভাল গাইছে। TF: *vAl gAICe* EF: Sang well.	V+Adj	গাইছে ভাল। TF: *gAICe vAl*
PP+V	যদি আহা ! TF: *yadi aAhA* EF: If (you) come?	V+PP	আহা যদি ! TF: *aAhA yadi*
PP+PN	তেনেহলে সি ! TF: *tenehle si* EF: Or else he!	PN+PP	সি তেনেহলে ! TF: *si tenehale*

Assamese has a number of morpho-syntactic characteristics which makes it different from other Indic language such as Hindi. Our study reveals that - word order at the clause level is free, and in some cases intra clause level ordering is also free that is elements which can be thought as a single semantics unit, can be reorder within the clause. The most favourite word order of Assamese is SOV. For example-

1) মই ভাত খালোঁ। (SOV)
   TF: *mai bhAt khAlo.*

EF: I ate rice.
Now we can arrange these 3 constituents in 3! Ways. Thus we get 6 possible combinations.

a) ভাত মই খালোঁ। (OSV) *bhAt mai khAlo.*
b) ভাত খালোঁ মই। (OVS) *bhAt khAlo mai.*
c) মই খালোঁ ভাত। (SVO) *mai khAlo bhAt.*
d) খালোঁ ভাত মই। (VOS) *khAlo bhAt mai.*
e) খালোঁ মই ভাত। (VSO) *khAlo mai bhAt.*

It is not necessary that all sentences have subject verb and object. For example in the following sentence verb is absent.

2) মই তেজপুৰ বিশ্ববিদ্যালয়ৰ ছাত্ৰ। (PN-N-N)
TF: *maI Tezpur-ViswavidyAlayor chAtra.*
ET: I am student of Tezpur University
In this case the verb হয় (equivalent to “ is ” in English) is absent and is a meaningful sentence. Though there are 4 words, তেজপুৰ বিশ্ববিদ্যালয় (ৰ) is a single constituent, a name of an university so number of constituent will be 3 and hence total of 3! grammatically correct combinations are possible. Let us consider another sentence-

3) মানুহজনে কুকুৰটো ৰাস্তাত দেখিছে।
TF: *mAnuhjane kukurTo rAstAt dekhise.*
ET: The man has seen the dog on the road.

NP—মানুহজনে (the man) (Man + Qnt: Single + Gender: Male + Vibhakti)
NP—কুকুৰটো (the dog) (dog + Qnt:Single + Gender: Neuter)
NP—ৰাস্তাত (on road) (road + Vibhakti)
VP—দেখিছে (saw) (see + past ind.)
Interesting property of such type of sentence is that we can simply exchange the position of noun phrase (NP) without changing the emphasis.

a) কুকুৰটো মানুহজনে ৰাস্তাত দেখিছে।
TF: *kukurTo mAnuhjane rAstat dekhise*
b) ৰাস্তাত মানুহজনে কুকুৰটো দেখিছে।
TF: *rAstAt mAnuhjane kukurTo dekhise*

If we put a numeral classifier এটা before NP কুকৰ then total number of constituent will be increased to 5, and the sentence will be-

4) মানুহজনে এটা কুকুৰ ৰাস্তাত দেখিছে।
TF: *mAnuhjane etA kukur rastat dekhise.*
EF: The man saw a dog on road.
In this case we will not get 5! numbers of grammatically correct combination. Because the count noun এটা(*etA*) modifies only কুকুৰ(*kukur*), not the others. Therefore during reordering of a sentence এটা কুকুৰ(*etA kukur*) is considered as a single constituent. Sometime within the constituent reordering of words are also possible. For example- এটা কুকুৰ(*etA kukur*) can be written as কুকুৰ এটা(*kukur etA*) without changing he meaning of the phrase. But from the sentence it will not be clear whether “The man saw a dog on road” or “The man saw dog on a road”.

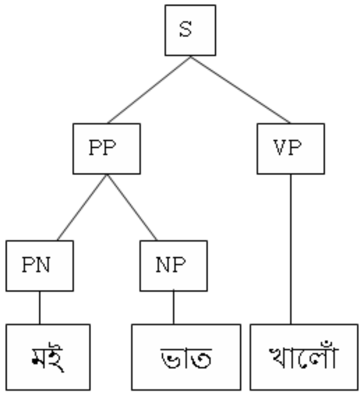

Fig. 1. Parse tree for sentence 1

a) মানুহজনে কুকুৰ এটা ৰাস্তাত দেখিছে।
TF: *mAnuhjane kukur etA rAstAt dekhise.*

5) আম মিঠা ফল।(N-ADJ-N)
TF: *aAm mithA phal.*
EF: Mango is fruit.
Here in this simple 3 constituent sentence if we try to exchange the position of noun(like example sentence 4) then we will get struturally correct but semantically wrong sentence.

a) ফল মিঠা আম.
TF: *phal mitha aAm*

Another important rule in this context is that the extension (Ext.) or the clauses as Ext. are always preceded by or followed by the constituent qualified. That is if element *A* is extension of *B* then *B* must be followed by *A* (in other words *A* does not occur after *B*). Consider the following example-

6) প্ৰধান শিক্ষকে আমাক সুন্দৰকৈ নতুন ব্যাকৰণ শিকাইছে ।
TF: *pradhAn sikhyake aAmak sundarkoi natun vyAkaran sikAIse*
EF: Head sir teaches us new grammar nicely.

প্ৰধান_Adj শিক্ষকে_N আমাক_PN সুন্দৰকৈ_Adv নতুন_Adj ব্যাকৰণ_N শিকাইছে_V

V. Parsing Assamese Sentences

As an initial exercise in parsing Assamese sentences, we present an approach for parsing simple sentences. We define a CFG grammar through which we can parse simple sentences like sentence (1) or any types of simple sentence where object is prior to verb. The parse tree of sentence (1) using the defined CFG grammar is shown in figure 1. In case of sentences 1(d) and 1(e) it generates a cross parse tree (Figure 2).

But unfortunately it can also generate a parse tree for sentence 5(a), which is semantically wrong. From sentence number 4 and 5 we can draw a conclusion that if the noun is attached with any type suffix, then it is easy for the defined CFG grammar to generate syntactically and semantically correct parse tree.

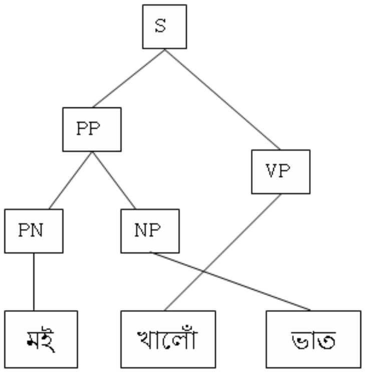

Fig. 2. Parse tree for sentence 1(d)

In Assamese two basic types of groupings of words are possible in a sentence. One is grouping adverb with verb and other is grouping adjective with noun. In general adverb or adjective occurs before the verb or noun respectively. Since Assamese is a relatively free word order language so these modifiers may occur anywhere in the sentence prior to verb or noun. It means that some constituent may occur in between adverb and verb or adjective and noun. In example sentence number 6, three types of grouping are possible- one verb group and two noun groups. Adjectives are adjacent to nouns but adverb occur prior to verb with a noun group in between. So after grouping we will get total 4 groups (Figure 3).

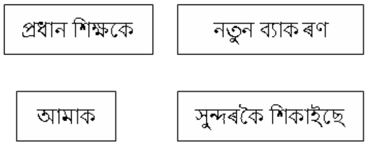

Fig. 3. Grouping of words of sentence 6

So we will get 4! grammatically correct sentences. But interestingly the main sentence from which the groups are formed is not included in this 4! combination. That is reordering the adverb again we can get another 6 new combinations. Though we mentioned above that adverb always occurs prior to verb, it is not always true. For example we can change the position of adverb and and verb within the group. That is সুন্দৰকৈ শিকাইছে can be reordered as শিকাইছে সুন্দৰকৈ. We can exchange the position of main object and subordinate object also. The constituent প্ৰধান শিক্ষক can be changed to শিক্ষক প্ৰধান. But here symbol of *Prathama Vibhakti* (Nominative case marker) এ is remove from S শিক্ষকে, and to the added to the Ext. of S. That is the new group will become শিক্ষক প্ৰধানে.

From figure 3 we can draw a complete graph considering each group as a vertex or node (Figure 4). A complete graph is a graph with all nodes are connected to each other. Now applying Chu-Liu-Edmond's maximum spanning tree algorithm we will obtain the parse tree for sentences which can not be obtained using our CFG grammar.

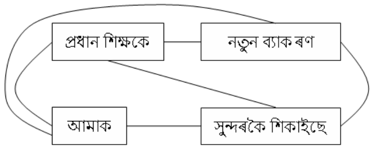

Fig. 4. Complete word graph

VI. Conclusion

Here we present the first step toward parsing of Assamese language. Our work is significant since Assamese has not received much attention of computational linguistic investigation. Using our approach we can handle simple sentences with multiple noun, adjective and adverb clauses. Handling of conjunction has been tackled to a limited extent. It needs to improved for complex sentences with different form. Also, there are other issues that we did not address in this paper.

References

[1] L. Tesniére, *Éléments de syntaxe structurelle*, Paris, Klincksieck, 1959.

[2] J. Bresnan and R. Kaplan, "Lexical-functional grammar: A formal system for grammatical representation," in *The Mental Representation of Grammatical Relations*, J. Bresnan, Ed., Cambridge, Massachusetts, 1982, MIT Press.

[3] A. K. Joshi, "An introduction to Tree Adjoining Grammar," *Mathematics of Language*, 1987.

[4] Carl Jesse Pollard and Ivan A. Sag, *Head-driven Phrase Structure Grammar*, University of Chicago Press, 1994.

[5] Akshar Bharati, Vineet Chaitanya, and Rajeev Sangal, *Natural Language Processing: A Paninian Perspective*, Prentice-Hall, India, 1993.

[6] Adwait Ratnaparkhi, Salim Roukos, and R. Todd Ward, "A maximum entropy model for parsing," in *In Proceedings of the International Conference on Spoken Language Processing*, 1994, pp. 803–806.

[7] Michael A. Covington, "A dependency parser for variable-word-order languages," Tech. Rep., The University of Georgia, 1990.

[8] Akshar Bharati, Samar Husain, Dipti Misra Sharma, and Rajeev Sangal, "A two-stage constraint based dependency parser for free word order languages," in *Proceedings of the COLIPS International Conference on Asian Language Processing 2008 (IALP*, Chiang Mai, Thailand, 2008.

[9] Terry Koo, Xavier Carreras, and Michael Collins, "Simple semi-supervised dependency parsing," in *Proceedings of ACL / HLT*, 2008.

[10] Joakim Nivre, Johan Hall, Sandra Kübler, Ryan McDonald, Jens Nilsson, Sebastian Riedel, and Deniz Yuret, "The conll 2007 shared task on dependency parsing," in *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007*, Prague, June 2007, p. 915932, Association for Computational Linguistics.

[11] Kenji Sagae and Alon Lavie, "A classifier-based parser with linear run-time complexity," in *Proceedings of the Ninth International Workshop on Parsing Technologies(IWPT),*. 2005, pp. 125–132, Association for Computational Linguistics.

[12] Joakim Nivre and Mario Scholz, "Deterministic dependency parsing of English text," in *Proceedings of COLING 2004*, Geneva, Switzerland, 2004, pp. 64–70.

[13] Hiroyasu Yamada and Yuji Matsumoto, "Statistical dependency analysis with support vector machines," in *Proceedings of the Ninth International Workshop on Parsing Technology*, 2003.

[14] Joakim Nivre and Jens Nilsson, "Pseudo-projective dependency parsing," in *Proceedings of the 43rd Annual Meeting of the ACL*, Ann Arbor, June 2005, pp. 99–106, Association for Computational Linguistics.

[15] Joakim Nivre, Johan Hall, and Jens Nilsson, "Memory-based dependency parsing," in *Proceedings of the Eighth Conference on Computational Natural Language Learning (CoNLL)*, Boston, Massachusetts, 2004, pp. 49–56.

[16] Yuchang Cheng, Masayuki Asahara, and Yuji Matsumoto, "Machine learning-based dependency analyser for Chinese," in *Proceedings of the International Conference on Chinese Computing (ICCC)*, 2005.

[17] Svetoslav Marinov and Joakim Nivre, "A data-driven dependency parser for Bulgarian," in *Proceedings of the Fourth Workshop on Treebanks and Linguistic Theories*, Barcelona, 2005, pp. 89–100.

[18] Gülşen Eryiğit and Kemal Oflazer, "Statistical dependency parsing of Turkish," in *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, Trento, 3-7 April 2006, pp. 89–96.

[19] Michael Collins, *Head-Driven Statistical Models for Natural Language Parsing*, Ph.D. thesis, 1999.

[20] Michael Collins, Lance Ramshaw, and Jan Hajič, "A statical parser for Czech," in *Proceedings of the 37th Annual Meeting - Association for Computational Linguistics*, 1999, pp. 505–512.

[21] Brooke Cowan and Michael Collins, "Morphology and reranking for the statistical parsing of spanish," in *HLT '05: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, Morristown, NJ, USA, 2005, pp. 795–802, Association for Computational Linguistics.

[22] Daniel M. Bikel, "Design of a multi-lingual, parallel-processing statistical parsing engine," in *Proceedings of the second international conference on Human Language Technology Research*, San Diego, california, 2002, pp. 178 –182, Morgan Kaufmann Publishers Inc. San Francisco, CA, USA.

[23] Amit Dubey and Frank Keller, "Probabilistic parsing for german using sister-head dependencies," July 2003.

[24] Ryan McDonald, Fernando Pereira, Kiril Ribarov, and Jan Hajič, "Non-projective dependency parsing using spanning tree algorithms," in *Human Language Technologies and Empirical Methods in Natural Language Processing (HLT-EMNLP)*, 2005.

[25] Ryan McDonald and Fernando Pereira, "Online learning of approximate dependency parsing algorithms," in *Proc. EACL-06*, 2006.

[26] Jason M. Eisner, "Three new probabilistic model for dependency parsing: An exploration," in *Proceedings of COLING-96*, 1996.

[27] Gosse Bouma and Gertjan van Noord, "Constraint-based categorial grammar," in *Annual Meeting - Association for Computational Linguistics*, 1994.

[28] Beryl Hoffman, "A CCG approach to free word order language," in *Proceedings of 30th annual meeting of ACL'02*, 1992.