Parsing for Natural Language in Odia :

A Novel Study

[1]Bishwa Ranjan Das,
[1]Department of Computer Science and InformationTechnology, Siksha 'O' Anusandhan Deemed to be University, Bhubaneswar, India
biswadas.bulu@gmail.com

[2]Dilip Singh,
[2]Department of Computer Science and Engineering, Siksha 'O' Anusandhan Deemed to be University, Bhubaneswar, India
dillipsingh0306@gmail.com

[3]Prakash Chandra Bhoi
[3]Department of Computer Science and InformationTechnology, Siksha 'O' Anusandhan Deemed to be University, Bhubaneswar, India
prakashbhoi@gmail.com

[4]Pusyanki Priyadarshini,
[4]Department of Computer Science and Engineering, Siksha 'O' Anusandhan Deemed to be University, Bhubaneswar, India
pusyanki.neha93@gmail.com

***Abstract*—This paper presents a simple parsing for Odia Language using Context-Free Grammar (CFG). Parsing is a method that is used to for building a sentence automatically in the phrases of grammar as well as in lexicon using syntactic analysis. It also includes semantic analysis and syntactic analysis, basically focusing on parsing for *Odia Language* based on *Context Free Grammar* followed by Panini method and top down approach. All the things are been represented as simple tree primarily formalism in context free grammar. Tokenization, POS Tagging, NP-Chunking and Morphological Analysis are been described in details of odia language.**

Keywords—Odia, Parsing, Panini, Tagging.

I. INTRODUCTION

The method for building a sentence automatically in the phrases of grammar as well as in lexicon using syntactic analysis which can be described by using the term *parsing*. The emerging syntactic analysis can be used as input to a system of semantic interpretation occasionally. Semantic analysis and syntactic analysis both can be included and can get the use of parsing.

The parsing output is something logically equivalent to a tree in the current linguistic formalism resulting part of sentence relation between dominance and precedence. In linguistic description the similar annotations of attribute-fee equations (*features*) is in the form of capturing other elements. There are many approaches for representing the distinct viable linguistic formalisms resulting in many distinct approaches for parsing to represent the consequences. To anticipate a simple tree representation, an implicit in context-free grammatical formalism. All the algorithms can be defined and can be used for more effective unification based formalisms to provide the preserve of context-loose *backbone* but in complexity and termination the properties may be different. Tailor-made grammar can be replaced with the training grammar with the help of parsing algorithms. The parsing algorithm has to have the several important properties if it could be used for practically. If it assigns an input of sentence it should do all the analysis concerning the current grammar and lexicon then it should be *complete*. The algorithm must be fully *efficient*, inevitable the negligible of computational work accurate with satisfying both the requirements and *robust*. Acts in a fairly pragmatic manner when presented with a sentence that it's far impotent to fully analyze successfully.

II. CONTEXT-FREE GRAMMAR

A context-free grammar is usually call them as phrases primarily based on the phrase that heads the constituent. In each production a single non-terminal symbol appear at left-hand side which is called context-free grammar (CFG). The most common manner of modeling the group of words behave as a phase. The concept of basing a grammar on constituent structure dates returned to Wilhem Wundt (1890), however now not formalized until Chomsky (1956) and independently through Backus (1959).

GM = <LT, NT, SS, RP>

- The set of terminals is LT (Lexicon).
- The set of non-terminals is NT for NLP, usually distinguish out a set P_N of pre-terminals which always rewrite as terminals.
- The start symbol is SS (the non-terminals)

- The form T is the rules/productions is RP, where T is a non-terminal.
- May be empty non-terminals.
- A grammar GM generates a language LG.

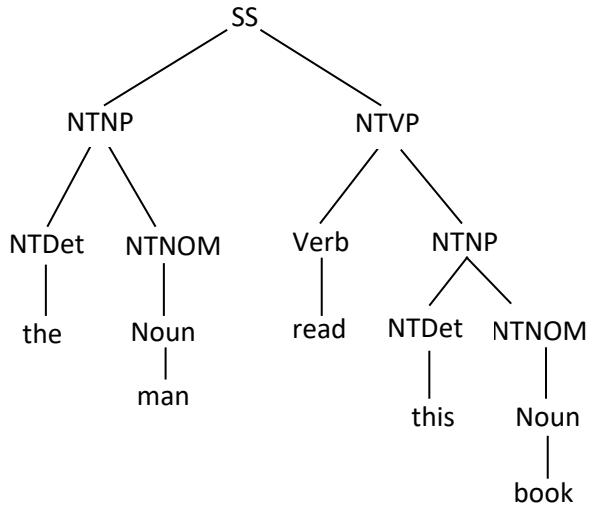
A. AN EXAMPLE CONTEXT-FREE GRAMMAR FOR ENGLISH LANGUAGE

GM = <LT, NT, SS, RP>

LT = {flight, read, the, does, that, man, this, a, meal, include, book}
NT = {NTS, NTNP, NTNOM, NTVP, NTDet, Noun, Verb, Auxi}
SS = SS
RP = {
NTS -> NTNP NTVP NTDet-> the | a | this | that
NTS -> Auxi NTNP NTVP Noun -> man | meal | flight | book
NTS -> NTVP Verb -> include | read | book
NTNP -> NTDet NTNOM Auxi -> does
NTNOM -> Noun
NTNOM -> Noun NTNOM
NTVP -> Verb
NTVP -> Verb NTNP
}

NTS -> NTNP NTVP
-> NTDet NTNOM NTVP
-> The NTNOM NTVP
-> NTVP the Noun
-> NTVP the man
-> Verb NTNP the man
-> NTNP the man read
-> NTDet NTNOM the man read
-> NTNOM the man read this
-> Noun the man read this
-> Complete : the man read this book

B. PARSE TREE FOR ENGLISH SENTENCE



(**Fig. 1** English sentence parsing)

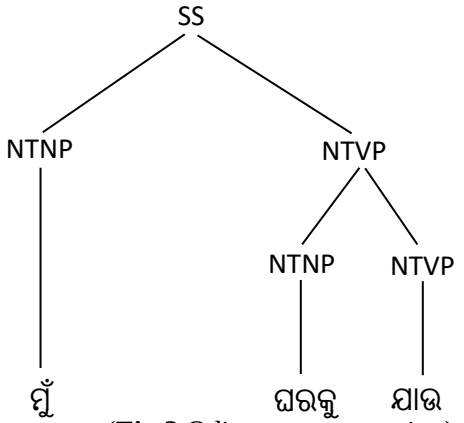
C. GRAMMATICALITY

A set of sentences is derived by the grammar can be defined as formal language in CFG, which is said to be grammatical. A sentence can be called as ungrammatical if it is outside of CFG as shown in Fig. 1.

D. TOP-DOWN PARSING

Top-down parsing is more suitable for implementation in a goal-directed language. A top-down parser begins with a list of constituents to be built. It rewrites the goals inside the goal listing with the list of matching one in increasing it with the RHS and opposition to the LHS of the grammar rules. Attempting the sentence to be derived which is to match.

An Example – ମୁଁ ଘରକୁ ଯାଉଛି



(**Fig.2** Odia sentence parsing)

This is in SOV format as shown in Fig. 2.

III. TOKENIZATION

When text data comes to work us mostly uses a common task which is known as tokenization. The entire text document converts to smaller unit or splitting a phase, paragraph or a sentence such as individual words or terms are been carried out by tokenization. These small segments or units are called as tokens. The process of demarcating and possibly classifying sections of a string of input characters.

An Example – ମୁଁ ଘରକୁ ଯାଉଛି(mu gharaku jauchhi). Here the word ମୁଁ, ଘରକୁ and ଯାଉଛି are each one token.

IV. POS TAGGING

The method for mapping a sequence of lexical categories from a sequence of words is done by POS Tagging. In each word of a sentence can be processed by allocating POS, like adverb, noun, pronoun, verb or other markers like lexical class. If a string of words of a natural language sentence is to be inputted to a tagging algorithm then the output of specific tag set for each word will be a single POS Tag.

An Example: ମୁଁ ଘରକୁ ବସରେ ଯାଉଛି(mu gharaku basre jauchhi).
Here the word ମୁଁ is noun, ଘରକୁ is noun, ବସରେ is noun and ଯାଉଛି is verb.

V. MORPHOLOGICAL ANALYSIS

Morphological analysis uses a method for a particular word to finding the root word and its lexical information including number, gender and person.

Fritz Zwicky has developed a method for non-quantified problem complex and discovering all the possible solutions to a multi-dimensional. In the surface form, every element of the lexicon is to be mapped to all possible roots, along with all possible lexical information. The possible parts of speech (PPOS) a surface form of the word that can be taken to obtain from Morphological Analysis from the point of POS tagging.

$$MA: LE \rightarrow 2^{PR*LI}$$

Where PR: - sets of all roots. LI: - set of all Lexical information.

An Example –

- ମୁଁ ଘରକୁ ଯାଉଛି(mu gharaku jauchhi)
- ମୁଁ - noun
- ଘର-root word – noun ଘରକୁ= ଘର+କୁ(vibhokti/ବିଭକ୍ତି-କୁ)
- ଯିବା-root word verb ଯାଉଛି = ଯିବା + ଉଛି

A. MORPHEMES

- A smallest meaning Unit in the grammar of language.
- A smallest linguistic unit that has semantic meaning.
- A smallest part of a word that can carry a discrete meaning.

An Example –

- The word "**Unbreakable**" has 3 parts/morphemes i.e. Un, break and able
- For Odia, ଆବାଳବ୍ରୁନ୍ଧବନିତା, In this word, three morphemes are there**,** ଆବାଳ, ବ୍ରୁନ୍ଧ**and** ବନିତା

VI. CONCLUSION AND FUTURE WORK

The method of tokenization is described for an Odia sentence, then POS Tagging is done for applying Morphological Analysis using Context-Free-Grammar. Top down parsing method also applied here. Next will try to a make a robust parser.

REFERENCES

Andrew McCallum, (2007). Context Free Grammars, "*Introduction to Natural Language Processing*", CS 585.

Dick Grune, Ceriel Jacobs, Parsing Techniques, "*A Practical Guide*"**,** Department of Mathematics and Computer Science, Vrije Universiteit, Amsterdam, Netherlands.

Aniket Dalal, Kumar Nagaraj, Uma Sawant, Sandeep Shelke, Pushpak Bhattacharyya **"***Building Feature Rich POS Tagger for Morphologically Rich Languages: Experiences in Hindi*", CSE department, IIT Bombay, Mumbai.

Bishwa Ranjan Das, (2015). et. al., "*Part of speech tagging in odia using support vector machine*", Procedia Computer Science, Vol - 48, Pages 507-512.

Itisree Jena, Sriram Chaudhury, Himani Chaudhry, Dipti M., (2011). "*Developing Oriya Morphological Analyzer Using Lt-toolbox*", Information Systems for Indian Languages Communications in Computer and Information Science Volume 139, pp 124-129

Pradipta Ranjan Ray, Harish V. Sudeshna Sarkar, Anupam Basu**,** "*Part of Speech Tagging and Local Word Grouping Techniques for Natural Language Parsing in Hindi*", Department of Computer Science & Engineering, Indian Institute of Technology, Kharagpur**,** INDIA 721302

S. Mohanty, P. K. Santi, K. P. Das Adhikari, (2005). "*Analysis and Design of Oriya Morphological Analyzer (OMA): Some Tests with OriNet*", Proceedings of symposium on Indian Morphology, Phonology and Language Engineering, IIT Kharagpur, India.

Dr. Dhaneswar Mahapatra, (2010). "*Adhunika Odia Byakarana*", 5th Edition, Kitab Mahal.

Russell & Norvig, Artificial Intelligence, "*A Modern Approach*", Pearson Prentice Hall.

Daniel Jurafsky & James H. Martin, (2011). "*Speech and Language Processing*", 4th Edition, Pearson.

Pandey, P., Khari, M., Kumar, R., & Le, D. N. (2018). "*Automatic Generation of Synsets for Wordnet of Hindi Language*". International Journal of Natural Computing Research (IJNCR), 7(2), 31-47.

Jain, A., Tayal, D. K., Khari, M., & Vij, S. (2016). "*A novel method for test path prioritization using centrality measures*". International Journal of Open Source Software and Processes (IJOSSP), 7(4), 19-38.