Shallow Parsing for Odia Sentences: An Approach

Bishwa ranjan Das
Computer Science & Engineering
Siksha 'O' Anusandhan (Deemed to be University)
Bhubaneswar, India

Rekhanjali Sahoo
Computer Science & Engineering
Einstein Academy of Technology & Management
Bhubaneswar, India

Brojo kishore Mishra
Computer Science & Engineering
GIET University
Bhubaneswar, India

***ABSTRACT*-This paper shows a shallow parsing for the language Odia using Context-Free-Grammar concept. Parsing is the method used to describe the process of automatically building syntactic analysis of a sentence in terms of given grammar and lexicon. Parsing is also used to include both syntactic and semantic analysis, basically we here focus on parsing for "Odia Language" based on the grammar i.e. "Context Free Grammar" followed by Panini method and top down approach is applied. Everything is assumed here as a simple tree representation and underlying Context free grammatical (CFG) formalism. In this paper most of part of Odia language is described details like Tokenization, POS Tagging,NP-chunking, and Morphological Analysis.**

Keywords: Odia, Parsing ,Panini, Tagging, Grammar.

I. INTRODUCTION

'Parsing' is the term used to describe the process of automatically building syntactic analyses of a sentence in terms of a given grammar and lexicon. The resulting syntactic analyses may be used as input to a process of semantic interpretation, occasionally; 'parsing' is also used to include both syntactic and semantic analysis. We use it in the more conservative sense here, however.

In most contemporary grammatical formalisms, the output of parsing is something logically equivalent to a tree, displaying dominance and precedence relations between constituents of a sentence, perhaps with further annotations in the form of attribute-value equations ('features') capturing other aspects of linguistic description. However, there are many different possible linguistic formalisms, and many ways of representing each of them, and hence many different ways of representing the results of parsing. Here a simple tree representation is assumed, and an underlying context-free grammatical (CFG) formalism. However, all of the algorithms described here can usually be used for more powerful unification based formalisms, provided these retain a context-free 'backbone', although in these cases their complexity and termination properties may be different. Parsing algorithms are usually designed for classes of grammar rather than tailored towards individual grammars. There are several important properties that a parsing algorithm should have if it is to be practically useful. It should also be 'complete'; that it, it should assign to an input sentence all the analysis it can have with respect to the current grammar and lexicon. Ideally, the algorithm should also be 'efficient', entailing the minimum of computational work consistent with fulfilling the first two requirements, and 'robust': be having in a reasonably sensible way when presented with a sentence that it is unable to fully analysis successfully. In this discussion

II. CONTEXT-FREE GRAMMAR

The most common way of modeling constituency.CFG = Context-Free Grammar = Phrase Structure Grammar= BNF = Backus-Naur Form. The idea of basing a grammar on constituent structure dates back to Wilhem Wundt (1890), but not formalized until Chomsky (1956), and, independently, by Backus (1959).

G = <T, N, S, R>

- T is set of terminals (lexicon)
- N is set of non-terminals For NLP, we usually distinguish out a set P_N of preterminals which always rewrite as terminals.
- S is start symbol (one of the non terminals)
- R is rules/productions of the form X, where X is a non terminal and is a sequence of terminals
- Non terminals (may be empty).
- A grammar G generates a language L.

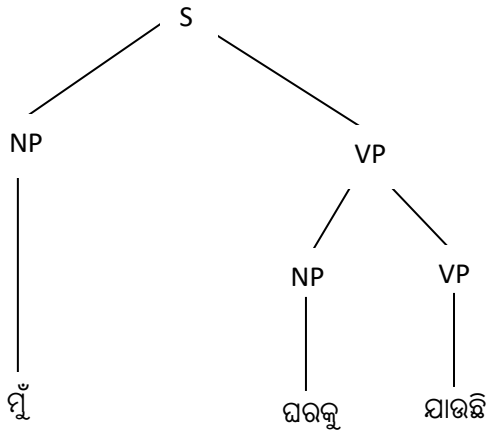
AN EXAMPLE CONTEXT-FREE GRAMMAR FOR ENGLISH LANGUAGE

G = <T, N, S, R>
T = {that, this, a, the, boy, book, flight, meal, include, reads, does}
N = {S, NP, NOM, VP, Det, Noun, Verb, Aux}
S = S
R = {
S -> NP VP Det-> that | this | a | the
S -> Aux NP VP Noun -> book | flight | meal | boy
S -> VP Verb -> book | include | reads
NP ->Det NOM Aux -> does
NOM -> Noun
NOM -> Noun NOM
VP -> Verb
VP -> Verb NP
}

S -> NP VP

- ✓ Det NOM VP
- ✓ The NOM VP
- ✓ The Noun VP
- ✓ The boy VP
- ✓ The boy Verb NP
- ✓ The boy reads NP
- ✓ The boy readsDet NOM
- ✓ The boy reads this NOM
- ✓ The boy reads this Noun
- ✓ The boy reads this book

PARSE TREE FOR ENGLISH SENTENCE



(Figure. 1 English sentence parsing)

A. GRAMMATICALITY

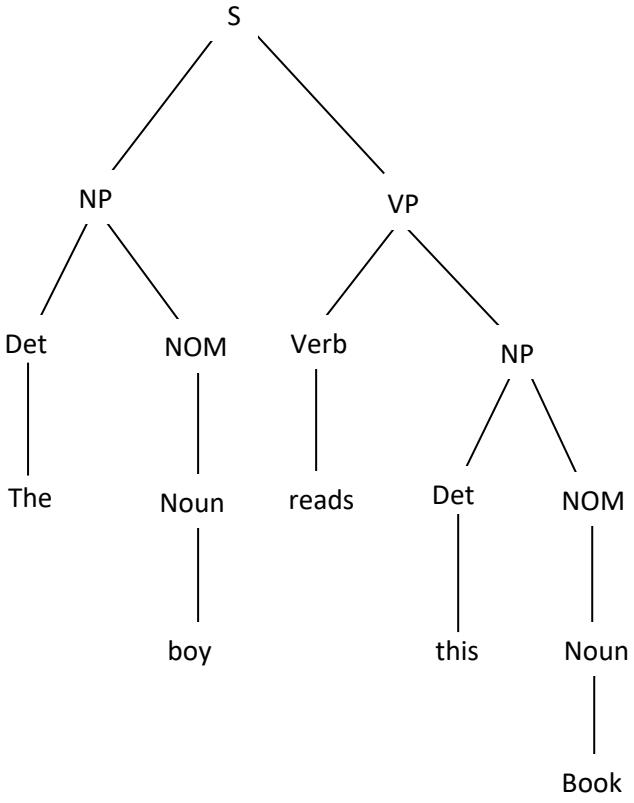
A CFG defines a formal language = the set of all sentences (strings of words) that can be derived by the grammar. Sentences in this set said to be grammatical. Sentences outside this set said to be ungrammatical as shown in Fig. 1.

B. TOP-DOWN PARSING

Top-down parsing is goal-directed.

- ❖ A top-down parser starts with a list of constituents to be built.
- ❖ It rewrites the goals in the goal list by matching one against the LHS of the grammar rules, and expanding it with the RHS
- ❖ Attempting to match the sentence to be derived.

Examples:-ମୁଁ ଘରକୁ ଯାଉଛି



(Figure.2 Odia sentence parsing)

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

C. TOKENIZATION

Tokenization is one of the most common tasks when it comes to working with text data. Tokenization is essentially splitting a phrase, sentence, paragraph, or an entire text document into smaller units, such as individual words or terms. Each of these smaller units are called tokens. The process of demarcating and possibly classifying sections of a string of input characters.

For Example:ମୁଁ ଘରକୁ ଯାଉଛି/mu gharaku jauchhi. Here the word ମୁଁ, ଘରକୁ and ଯାଉଛି are each one token.

D. POS TAGGING

POS Tagging is mapping from a sequence of words to a sequence of lexical categories.POS Tagging is the process of assigning a part of speech, like noun, verb, pronoun, adverb, adverb or other lexical class marker to each word in a sentence. The input to a tagging algorithm is a string of words of a natural language sentence and a specific tag set the output is a single POS Tag for each word.

For Example: ମୁଁଘରକୁବସରେ ଯାଉଛି/mu gharakubasrejauchhi.

Here the word ମୁଁis noun, ଘରକୁis noun, ବସରେ is noun and ଯାଉଛିis verb.

E. MORPHOLOGICAL ANALYSIS

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

It is a method developed by Fritz Zwicky (1967, 1969) for exploring all the possible solutions to a multi-dimensional, non-quantified problem complex. It is a mapping from every element of the lexicon to all possible roots, along with all possible lexical information about the root that can be obtained from the surface form. From the point of POS tagging, the possible parts of speech (PPOS) a surface form of the word can take may be obtained from Morphological Analysis itself.

$$MA : L \rightarrow 2^{R*I}$$

Where R :- all possible roots. I :- set of all possible Lexical information

Examples:-

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

III. CONCLUSION AND FUTURE WORK

Here the method of tokenization is described for an Odia sentence, then POS Tagging is done applying Morphological Analysis using Context-Free-Grammar. Top down parsing method also described here for avoid confusion in syntactic level. Next a robust and deep parser will be made for Odia language using computational intelligent method by the system.

REFERENCES

[1]. Context Free Grammars, “Introduction to Natural Language Processing”, CS 585, Fall 2007, Andrew McCallum.

[2]. PARSING TECHNIQUES A Practical Guide**,** **“**DICK GRUNE, CERIEL JACOBS”,Department of Mathematics and Computer Science, VrijeUniversiteit, Amsterdam, Netherlands.

[3]. AniketDalal,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.

[4]. BishwaRanjan Das, et. al., “Part of speech tagging in odia using support vector machine”, Procedia Computer Science, Vol - 48, Pages 507-512, 2015

[5]. Itisree Jena, SriramChaudhury, Himani Chaudhry, Dipti M., “Developing Oriya Morphological Analyzer Using Lt-toolbox”, Information Systems for Indian Languages Communications in Computer and Information Science Volume 139, 2011, pp 124-129

[6]. PradiptaRanjan Ray, Harish V. SudeshnaSarkar,AnupamBasu**,** “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

[7]. S.Mohanty, P.K. Santi, K.P. Das Adhikari, “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, 2005.

[8]. AdhunikaOdiaByakaranaby Dr. DhaneswarMahapatra, KitabMahal, 5th Edition, 2010.

[9]. Artificial Intelligence, “A Modern Approach by Russell &Norvig”, Pearson Prentice Hall.

[10]. Speech and Language Processing by Daniel Jurafsky& James H. Martin, Pearson.