

# Analysis and Design of Oriya Morphological Analyser : Some Tests with OriNet

Sanghamitra Mohanty  
[sangham1@rediffmail.com](mailto:sangham1@rediffmail.com)

Prabhat Kumar Santi  
[pksanti@rediffmail.com](mailto:pksanti@rediffmail.com)

K.P. Das Adhikary  
[krushnapada@rediffmail.com](mailto:krushnapada@rediffmail.com)

PG Department of Comp. Sc. and Application  
Utkal University  
Bhubaneswar, Orissa, India- 751004.

## Abstract

Indian languages are characterised by a rich system of inflections (VIBHAKTI), derivation and compound formation for which a standard Morphological Analyser (MA) is needed for a machine to deal with the lexicons of those languages [2]. The numbers of word have been derived from the root word by some specific orthographic rule in the Indian languages. This paper deals with the analysis and design of Oriya Morphological Analyser (OMA). The lexical category and computational grammatical model of Oriya language are described to build a complete OMA in the current scenario. The major contents on which our OMA has been built up with i) Pronoun Morphology (PM), ii) Inflectional Morphology (IM) and iii) Derivational Morphology (DM) [7]. The OMA system is designed according to the object oriented approach to increase its reusability, robustness and extensibility. We have developed and implemented the Decision Tree (DT) and its respective algorithm of each type of morphology, through which our OMA runs successfully. Some tests of the OMA with OriNet: the Word-Net for Oriya Language has been reflected in this paper.

The present work aims to build a computational model for the analysis and generation of Oriya language, the language being one of the official languages of the state of Orissa, situated in the eastern part of India. An important advancement of computerized language generation and understanding is to capacitate Morphological Analysis (MA) for any type of lexicon. For example, Oriya words like, ବାଳକମାନେ (*bAlakamAne*) (boys), the system technically reduces it to root word ବାଳକ (*bAlaka*) (boy) and at the same time provides other information like part-of speech (i.e., noun), suffix (ମାନେ), gender (i.e., masculine), number (i.e., plural) and case-ending relation (i.e., subjectivity). This analysis is performed by our OMA, which not only deals with the study of words but also its morphemes. A morpheme is

defined as the minimal meaning-bearing unit in a language. Morphemes vary from language to language and when it is added to the root word, denote different meaning. It can be divided into three types such as prefix, suffix and infix. Prefix and suffix precede root word whereas infix gets inserted inside the root word. A word can be easily identified by the system as a sequence of characters delimited by spaces, punctuation marks. A word can be of two types as simple and compound. A simple word consists of a root or stem together with suffixes or prefixes. A compound word can be broken into two or more simple words.

The OMA system is needed for various applications in developing NLP tools [1,3,5]. For example, in OMT, there is need of root words, which is obtained through the OMA. In the words like ଯାଇଛି (*jAichhi*) (has/have gone), ଯାଇଥିଲି (*jAithiLi*) (had gone) etc., OMA provides both the information of root word as ଯା (*jA*)(go) and suffix as ଈଛି (*ichhi*) and ଈଥିଲି (*uthili*) to the OMT. Similarly, it has typical use in OriNet for searching any type of lexicon.

In the process of forming words in Oriya language there exists three major classes of morpheme such as [7] Pronoun Morphology (PM), Inflectional Morphology (IM), Derivational Morphology (DM)

The PM is the study on the grammatical classification of pronoun. For example the pronoun ତୁମ୍ଭମାନେ (*tumbhemAne*) (you) indicates that it is personal pronoun, 2<sup>nd</sup> person and plural number. The PM of Oriya language has been classified into different groups according to their syntactic rules. Some of the pronouns are ambiguous as in the interrogative pronouns like କିଏ (*kie*) (who) is used as plural as well as singular. For these ambiguities, further modification is necessary.

The IM is the combination of root word with grammatical morphemes, usually resulting in a word of the same class as the original stem and filling some syntactic function like agreement. In other words, the morpheme, which conveys 'number',

‘person’, ‘inflection’, ‘*kAraka*’ etc., is called IM. For example the inflectional morphemes ମାଣେ (*mAne*) for making plural form (number), third person (person), subject (*kAraka*) and 1<sup>st</sup> inflection (inflection) etc., in case of nominal word. Similarly, from the inflectional morphemes like ଉଥାଣ୍ଟି (*uthAnti*), we obtain information as present tense, 1<sup>st</sup> person and singular number in case of verbal word. Like Sanskrit, Oriya language has also strong inflectional system, which can be classified into two classes such as nominal morphemes and verbal morphemes. There are nearly 40 morphemes for nominal words and 100 morphemes for verbal words in Oriya. The nominal morphemes are attached to nominal words, whereas verbal morphemes are attached to verbal words. All the suffixes in IM determine the number, person, case-ending relation, tense and inflection of the word. The verbal morphemes are quite complicated than nominal, which are classified according to the tense, person, and number attached to the verbal word only. Moreover, there are also some ambiguous morphemes in both the cases of nominal and verbal as in the nominal morpheme ଳ୍ଵ (*Ngku*), which in one hand indicates 2<sup>nd</sup> inflectional singular number on the other hand 4<sup>th</sup> inflectional singular and plural number. But, our Morphological Parser (MP) successfully handles them and provides different alternatives too.

The DM is the combination of a word stem (root word) with a grammatical morpheme, usually resulting in a word of different class, often with a meaning hard to predict exactly. The prime characteristic of this morphology is that even if it doesn't imply any number, person etc., like IM, but has an important role to convert from one lexical category to other. For example a verb word ହସ୍ (*Hasx*) (laugh) can take the derivational suffix i.e., ଏଇବା (*eibA*) to produce a nominal word as ହସେଇବା (*HaseibA*) (making laugh). Similarly, also a noun word (ANThu) is converted to a verb word as (ANTheibA) by use of DM i.e., ଏଇବା (*eibA*). The DM is quite complex but it is handled easily by OMA. There are so many words belonging to verb group derived from nominal words and also words belonging to verb group derived from verbal words. Generally, we find two types of DM such as prefix and numberless morphology in Oriya. The prefix can be further classified into two groups such as modified prefix (modifies the meaning of the root word) and negative prefix (changes to the antonym of the root word). Similarly, the numberless morphemes can be classified into five groups such as VsN (Verbal Noun), VsV (Verbal Verb), Gs (Gender suffix), NsV (Nominal Verb) and NsN (Nominal Noun) [4,7]. For examples, the word ପରାଜୟା (*parAjaya*)

(defeat) is derived from the negative prefix ପରା (*para*) and the word ଜୟା (*jaya*) (win).

The architecture of OMA is divided into five parts (Figure-1) such as OriNet Database (OD), which stores the Oriya lexicon (Only root words) whereas OMA Engine (OE) processes the system and Morphological Parsing (MP) parses the word according to orthographic rule. Decision Tree (DT) decides to classify the morphemes and different functional programmes by use of OMA[6].

The OMA system has been designed on the basis of Object-Oriented Approach (OOA). By use of this design methodology, different functions can be added or deleted to the existing system conveniently. We have implemented Pronoun Morphology and Inflection Morphology in the OMA in such a manner that it is successfully run with the OriNet system (Figure-2), Oriya Spell Checker (OSC) and Oriya Grammar Checker (OGC) [1,3,5]. The OSC handles any type of word (derived, inflectional or root) using the OMA

It also provides sufficient interface for applications involved in Oriya Machine Translation (OMT), Word-Net for Oriya (OriNet), Oriya Spell Checker (OSC) and Oriya Grammar Checker (OGC) [4,5]. All these developments have been worked out on the basis of the syntactic approach of Sanskrit language for which, we hope the technology involved here can be extended to any other Indian languages.

**Keywords:** Word-Net, OriNet, Morpheme, Inflectional Morphology, Derivational Morphology and Pronoun Morphology

## Reference

- [1] Alam. Y. S. et. al. “*Lexicons in an Object-Oriented Grammatical Model For Universal Grammar – Based Machine Translation (UGMT)*”. Proceedings of the 1<sup>st</sup> GWN conference January 21-25, 2002, CIIL, Mysore.
- [2] Bharati. A. et. al. (1995) “*Natural Language Processing, A Paninian Perspective*”. Prentice Hall of India Private Limited, New Delhi-110001.
- [3] Chadhuri B. B. et. al. “*To wards Indian Language Spell Checker Design*” IEEE Proceedings of LEC-2002, University of Hyderabad, Hyderabad, India.
- [4] Fellbaum C. Editor (1999). “*WordNet: An Electronic Lexical database*”. The MIT Press, Cambridge, Massachusetts, London, England.
- [5] Mohanty S et al “*Object Oriented Design Approach to OriNet System: On-line Lexical Database for Oriya Language*”. IEEE Proceedings of

LEC-2002, University of Hyderabad, Hyderabad, India.

[6] Rayner D' Souza et al. "Natural Language Generation from Semantic Net like Structures with application to Hindi", STRANS- 2001, IIT Kanpur, Kanpur, India.

[7] Mohapatra Pandit N. and Dash S.(2000). "Sarbasara Byakarana", New Students Store, Cuttack, Orissa, India

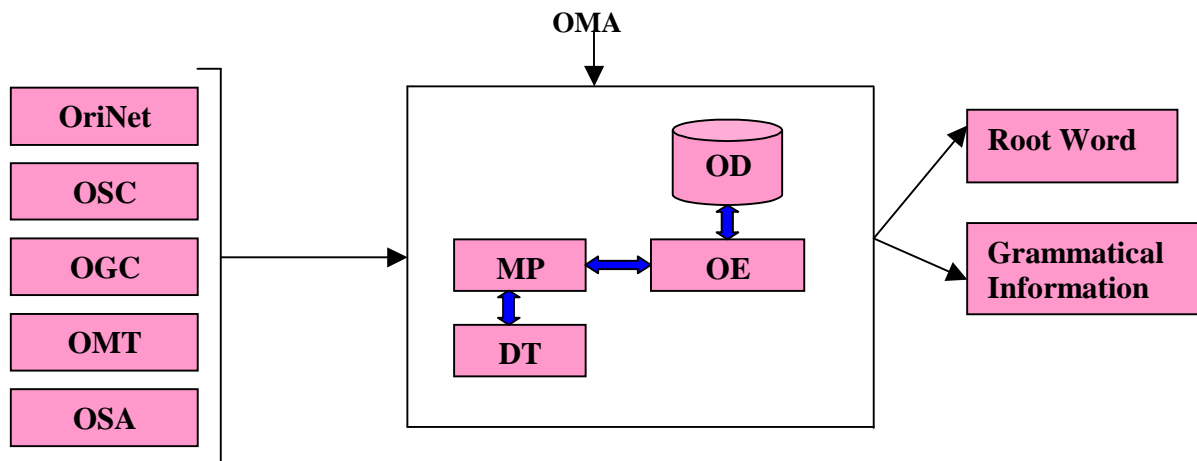


Figure -1 Architecture of the OMA System.

(OD = OriNet Database, OE = OMA Engine, MP = Morphological Parsing, DT = Decision Tree, OriNet = WordNet for Oriya, OSC = Oriya Spell Checker, OGC = Oriya Grammar Checker, OMT = Oriya Machine Translation and OSA = Oriya Semantic Analysis)

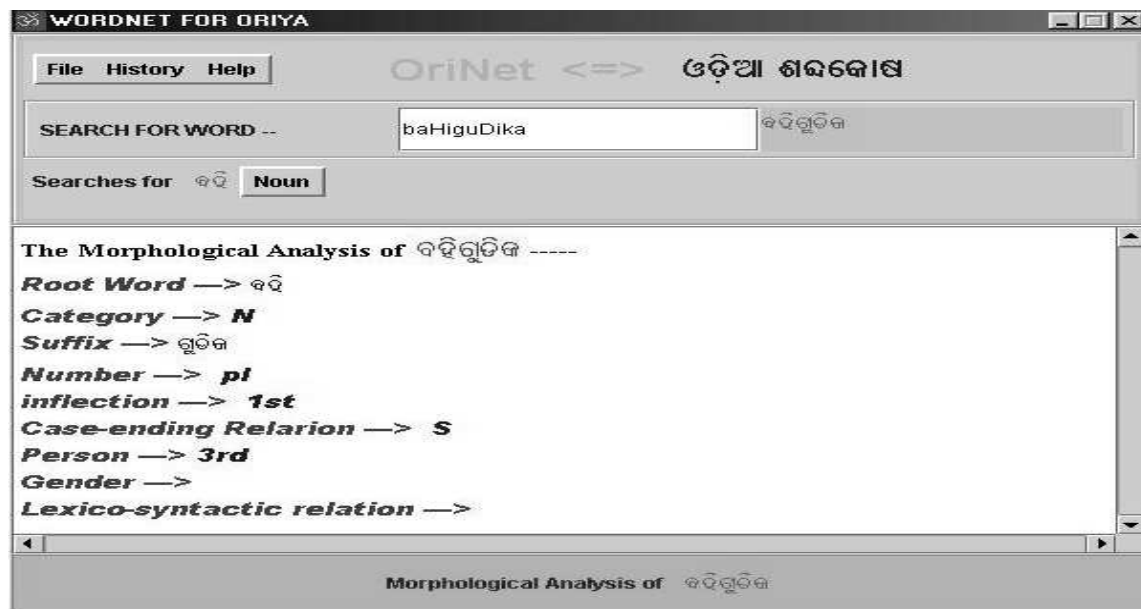


Figure 2:- Output of the OMA of the word ବାହିଗୁଡ଼ିକ "baHiguDika" (books) in OriNet. (N = Noun, Pl = Plural, 1<sup>st</sup> = 1<sup>st</sup> inflection, S = Subject, 3<sup>rd</sup> = 3<sup>rd</sup> person)