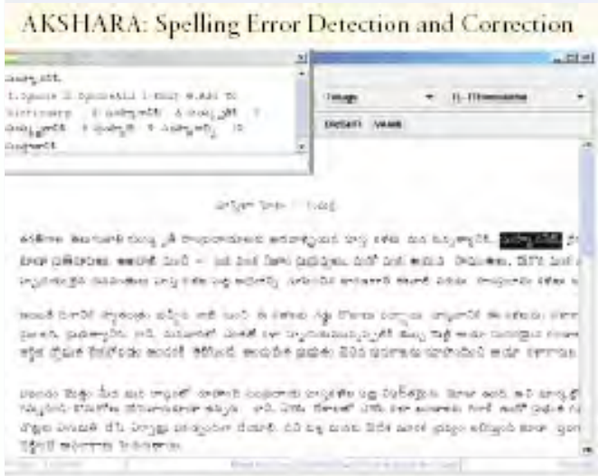
HMM models have also been built. These models can be used to detect spelling errors and to rank the suggestions for correcting a given word.

- **Representative Screenshot**



- **Operating System used :** LINUX
- **Scalability / Portability / Expandability :** Available
- **Availability of documentation:** Yes
- **Testing of the Product / Technology :** Testing has been done
- **IPR / Open Source :** IPR lies with University of Hyderabad and Dept of Information Technology.
- **Potentical beneficiaries :** Telugu word processing software developers.
- **User agency tie-up :** Negotiation with Modular Infotech.
- **Name and address of the Resource Person :**

Dr. K. Narayanamoorthy
Department of CIS, University of Hyderabad
Hyderabad-500046
A.P. India
E-mail : knmcs@uohyd.ernet.in
Web : http://www.languagetechnology.ac.in

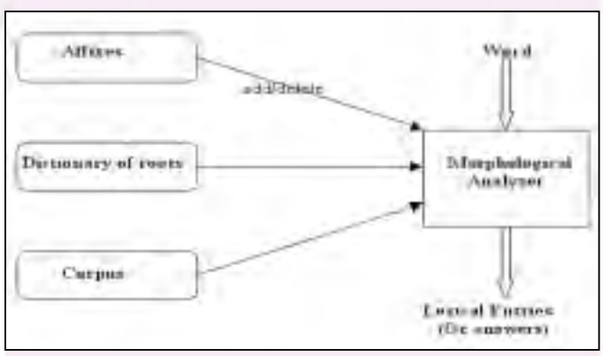
5.2 Morph Analyzer

5.2.1

- **Name of the Technology :** ***Assamese and Manipuri Morphological Analyzers.***
- **Nature of Technology** : Knowledge tool (Aid to machine translation system and Spell Checker.)
- **Level: (Product / Technology / Sub-system):** Technology
- **Technical Description of the Technology / Product including Basic block diagram, Algorithm used, O/S used, Front-end / user interface, and Specification of the Technology / Product:**
- **Assamese Morphological Analyzer:** The analyzer has been developed for use with the Spell checker and the Machine translation systems.

  Technical Description Stemming technique forms the base of the Assamese morphological analyzer. In the above technique, affixes are added/deleted according to the linguistic rules. The derived words are verified with the existing Corpus/Dictionary to treat as valid words.

  Block Diagram



Features

a) Currently works with Twenty linguistic rules

b) More rules can be added without alteration in code.

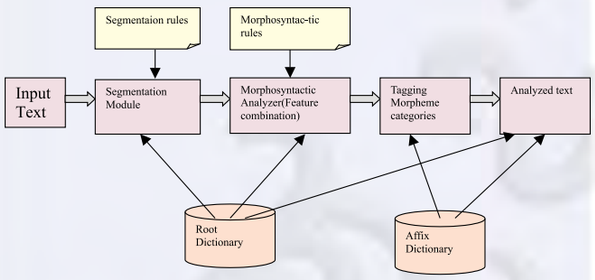
c) Modules are available in the form of API's for customization.

- **Manipuri Morphological Analyzer:** A Morphological Analyzer for Manipuri has been designed and implemented. The morphological processing is based on the grammatical rules and the dictionaries: root and affix dictionary. A root dictionary containing 3000 root words and an affix dictionary containing 55 affixes are being used. Identification of linguistic rules for nouns and pronouns has also commenced. Further linguistic rules for other grammatical categories are being studied.

Technical Description

The Manipuri Morphological Analyzer is also using the stemming technique as it is used in Assamese Morphological Analyzer. A model tagger is added to tag the analyze word. The tagger tags the lexical category of the root and the grammatical category of the affixes. The code is written using Perl script and the user interface has been developed using Perl/Tk. The dictionaries are stored in MS-Access database.

Block Diagram



Algorithm:

Step 1: Read the input text to be analyzed.

Step 2: If the text consists of more than one words then break the input text into

words. n<=No. of words.

Step 3: for i=1 to n

Check the i^{th} word in the root dictionary for a whole match.

If match is found then

"the input word is the root".

Else, call the segmentation module to segment the word into root and affix.

If affix is suffix then call separate morphemes to isolate each morpheme from the suffix.

Call the check_ grammar module for checking the morpho syntactic features.

If the morpho syntax is correct then call the tagging function to tag the categories.

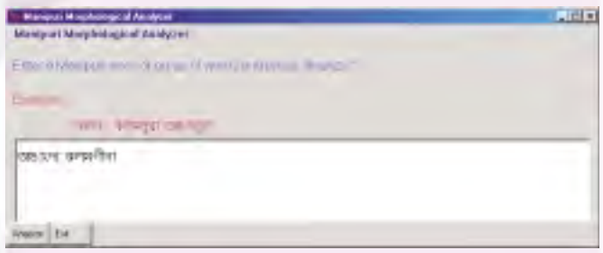
next i.

Step 4: End of the procedure.

Specifications

- Forty rules are being used currently
- Can be easily upgraded for more complex patterns.
- The modules, which have been developed so far are available in the form of API's and can be used according to the need by other applications like, spell checker, machine translation, etc.

- **Representative Snapshot / screenshot of the Technology / Product:**



- **Scalability / Portability / Expandability**: Scalable and Portable.
- **Readiness of Transfer of Technology (ToT)**: Preliminary documentation is ready.
- **Availability of documentation**: Preliminary documentation is ready.
- **Testing of the Product / Technology** : In progress
- **IPR / Open-source**: Open Source.
- **Potential beneficiaries**: Research institutions.

- **User-agency tie-up:** None
- **Name and address of the Resource Person:**
S.Choudhury, si_chow@iitg.ernet
S.Borgohain, samir@iitg.ernet.in
S.B.Nair, sbnair@iitg.ernet.in
P.K.Das, pkdas@iitg.ernet.in
Department of Computer Science & Engineering, Indian Institute of Technology Guwahati Assam – 781039 (India)
Web : http://www.iitg.ernet.in/rcilts

5.2.2

- **Name of the Technology:** ***Oriya Morphological Analyser (OMA)***
- **Nature of the Technology :** Knowledge tool (Application and Research oriented)
- **Level:** (Product / Technology / Sub-system): Product
- **Technical Description of the Technology / Product including Basic block diagram, Algorithm used, O/S used, Front-end / user interface, and Specification of the Technology / Product:** Indian languages are characterised by a very rich system of inflections (VIBHAKTI), derivation and compound word formation for which a standard morphological analyser is needed to deal with any type of text. The number of words are being derived from a given root word by some specific syntactic rules. Our Oriya Morphological Analyser (OMA) deals with Pronoun Morphology (PM), Inflectional Morphology (IM) and Derivational Morphology (DM). We have developed and implemented the Decision Tree (DT) and its respective algorithm for each type of morphology, to run our OMA successfully. It also provides sufficient underlying interfaces for applications involving Oriya Machine Translation (OMT), WordNet for Oriya (OriNet), Oriya Spell Checker (OSC) and Oriya Grammar Checker (OGC). All these development have been worked out on the basis of the syntactic periphery of Sanskrit language for which, we hope the technology involved here can be extended to any other Indian languages.
- **Representative Snapshot / screenshot of the Technology / Product:**

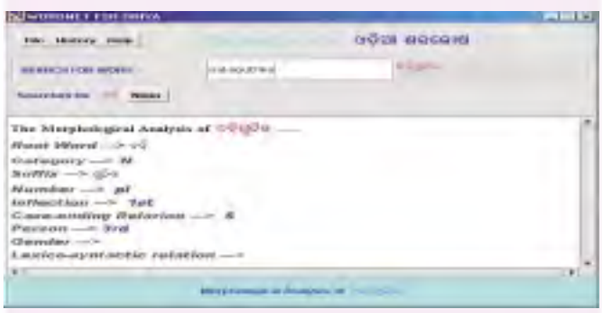


Figure : Out put of the OMA for the Oriya derived word "baHiguDika" in OriNet.

(N = Noun, Pl = Plural , 1st = 1st inflection, S = Subject, 3rd = 3rd person)

- **Scalability / Portability / Expandability:** All
- **Readiness of Transfer of Technology (ToT):** Yes
- **Availability of documentation :** Yes
- **Testing of the Product / Technology :** Under progress
- **IPR / Open-source:** Not applied
- **Potential beneficiaries:** Common man
- **User-agency tie-up:** Solicited
- **Name and address of the Resource Person:**
  Dr (Ms) Sanghamitra Mohanty
  RC-ILTS-Oriya, Dept. of CSA,
  Utkal University,
  Bhubaneswar, Orissa.

5.2.3

- **Name of the Technology :** ***Tamil Morph Analyzer (Atcharam)***
- **Nature of Technology:** Linguistic tool
- **Level :** Sub-system
- **Technical Description of the Technology**

Description

Tamil is a morphologically rich language. The root words combine with the morphemes in the form of suffixes. The Morphological analyser takes a derived word as input and separates it into root word and associated morphemes.

It is the basic tool used in spell checker, grammar checker, parser and machine translation systems. It is also used for extracting the root words from inflections in Tamil Search Engine and Online Tamil Dictionary.

Features

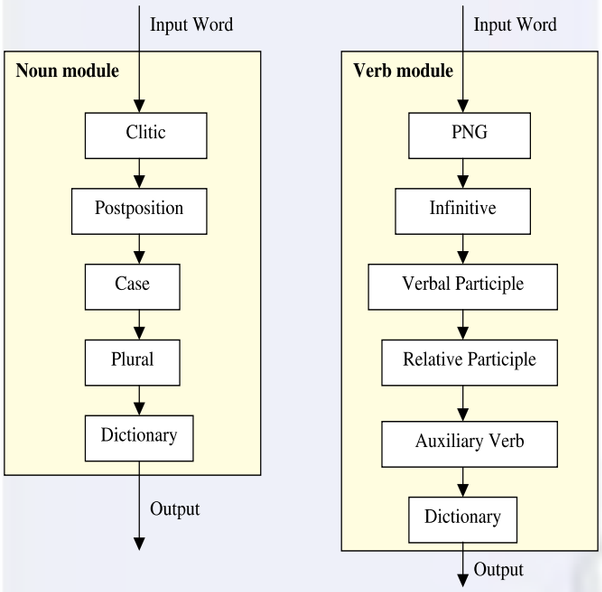
1. It uses a dictionary of 15,000 root words classified on fifteen categories.
2. It has two major modules – noun analyser and verb analyser.
3. The Verb module handles verb inflections like tense markers, person-number-gender markers, auxiliary verbs and clitics.
4. The Noun module handles noun inflections like plural markers, case markers, postpositions and clitics.

Algorithm

1. Given an input string, the processing starts from right to left to look for suffixes. A list of suffixes is maintained.
2. The suffixes are compared with the suffix list and the longest match is identified.
3. It removes the last suffix, and its tag is determined and added to the word's suffix list.
4. The remaining part of the word is checked in the dictionary and if found the process is

exited. The search in the dictionary is carried out using "BTree method".

5. According to the identified suffix, the next possible suffix list is generated.



The steps 2 to 5 are repeated.

Basic block diagram of the system

User Interface

Input the Tamil word to be analysed in the 'input' textbox (provided at the top) and click the button "Analyse". The result will be displayed in the 'output' textbox.

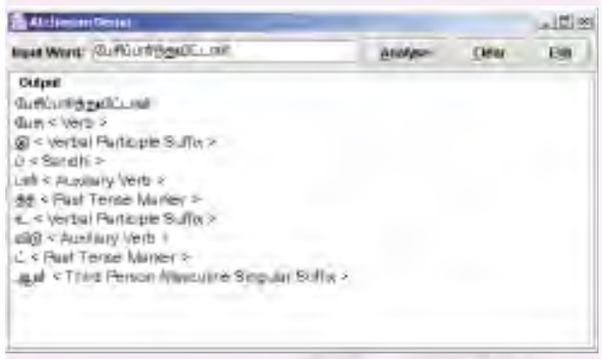
Specifications of Technology: Software components

The system was developed using Java on Windows 2000 platform.

Minimum requirement for the tool

1. J2re1.4.0
2. Any Tamil Keyboard driver(Font Encoding in TAB) to type in Tamil
3. Code-converter for other encodings is provided.
4. Operating System: Windows/Linux

Snapshot:



- **Scalability/Expandability** : Yes. More morphological rules can be added. Dictionary size can be increased.
- **Portability**: Yes.
- **Readiness of Transfer of Technology**: Yes
- **Availability of documentation**: User and Technical manuals are available.
- **Testing of the Product / Technology**: Tested with a file from CIIL Corpus. The result is 75% approximately.
- **IPR / Open-source**: IPR
- **Potential beneficiaries**: Linguist
- **User-agency tie-up**: RCILTS-Tamil has tied up with Modular Infotech, Pune
- **Name and address of the Resource Person:**
Dr.T.V.Geetha and Dr. Ranjani Parthasarathi
RCILTS Tamil, Anna University, Chennai.
E-mail : rp@annauniv.edu
Web : http://ns.annauniv.edu