IJISET - International Journal of Innovative Science, Engineering & Technology, Vol. 1 Issue 5, July 2014.
www.ijiset.com
ISSN 2348 – 7968

Morphological Analyzer for Classical Tamil Texts: A Rule-based approach

R.Akilan* and Prof. E.R.Naganathan#

*(Research Scholar, Department of Computer Science, Bharathiar University, Coimbatore)
Programmer, Central Institute of Classical Tamil, Chennai.
akilan.rp@gmail.com

#Head of Department, Department of Computer Science and Engineering, Hindustan University, Chennai. ernindia@gmail.com

Abstract

Morphology is the identification, analysis, and description of the structure of a given language's morphemes and other linguistic units, such as root words, affixes, parts of speech, intonations and stresses, or implied context. Morphological analysis is a process of segmenting words into morphemes and a process of analyzing the word formation. Morphological Analyzer is a program which takes words as input and produces its grammatical structure as output. It identifies and segments the words and assigns the grammatical information. Capturing the agglutinative structure of Classical Tamil words by an automatic system is a challenging job. This paper deals with how the Morphological Analyzer is developed for Classical Tamil texts using rule-based approach.

Keywords: Morphological Analyzer, Classical Tamil, Natural Language Processing, Morphological Analyzer for Classical Tamil.

I. INTRODUCTION

Natural language processing (NLP) is a field of computer science, artificial intelligence, and linguistics concerned with the interactions between computers and human (natural) languages. As such, NLP is related to the area of human computer interaction [1]. Many challenges in NLP involve natural language understanding, that is, enabling computers to derive meaning from human or natural language input, and others involve natural language generation.

II. MORPHOLOGICAL ANALYSIS

Morphology is a branch of linguistics that deals with word formation, analysis and generation. It is concerned with how words are formed or created in a language from smaller units in a systematic way. It has to find as well as to describe the mechanism behind this process. Each language has morphemes which are its smallest meaningful units. Realization of morphemes as part of a word is called morph. In formation of words from these smaller units, certain processes such as inflection, derivation and compounding etc., are involved. Also involved is morph tactics that determines how morphs should be put together to form words.

III. COMPUTATIONAL MORPHOLOGY

Computational Morphology is application of morphological rules in the field of computational linguistics. Computational Linguistics is an emerging area in Artificial Intelligence that is related to accomplish linguistic tasks through computational methods. Computational Morphology deals with the processing of words in both their grapheme and phonemic forms. Its most basic task can be defined as taking a string of characters or phonemes as input and delivering an analysis as output. It has various practical applications from low-level applications to speech and language processing systems [1].

IV. CLASSICAL TAMIL

Tamil literature has a rich and long literary tradition spanning more than two thousand years. The oldest extant works show signs of maturity indicating an even longer period of evolution. Contributors to the Tamil literature are mainly from Tamil people from South India, including the land now comprising Tamil Nadu, kerala, Sri Lankan Tamils from Sri Lanka, and from Tamil diaspora [2]. Also, there have been notable contributions from European authors. The history of Tamil literature follows the history of Tamil Nadu, closely following the social, political and cultural trends of various periods. The early Classical Tamil literature, starting from the period of 2^{nd} century BC, contain anthologies of various poets dealing with many aspects of life, including love, war, social values and religion.

ISSN 2348 – 7968

V. CLASSICAL TAMIL MORPHOLOGY

Classical Tamil Morphology is very rich. It is an agglutinative language, like the other Dravidian languages. Classical Tamil words are made up of lexical roots followed by one or more affixes. The lexical roots and the affixes are the smallest meaningful units and are called morphemes [3]. Classical Tamil words are therefore made up of morphemes concatenated to one another in a series. The first one in the construction is always a lexical morpheme (lexical root). This may or may not be followed by other functional or grammatical morphemes [3]. The morphological structure of Classical Tamil is quite complex since it inflects to person, gender, and number markings and also combines with auxiliaries that indicate aspect, mood, causation, attitude etc in verbs. A single verb root can inflect for more than two-thousand word forms including auxiliaries. Noun root inflects with plural, oblique, case, postpositions and clitics. A single noun root can inflect for more than five hundred word forms including postpositions. The root and morphemes have to be identified and tagged for further language processing at word level.

VI. MORPHOLOGY IN INDIAN LANGUAGES

Morphological analyzer is an integral part of any Natural Language Processing system, especially in the context of Indian languages where morphology plays significant role due to high inflectional and derivational nature of these languages. For fixed word-order languages, the semantics of a word are primarily governed by its absolute and relative position within a sentence. But for free word-order languages, the position of words in the sentence cannot provide much clue about its semantics. As in the case of Indian languages, which are mostly free order, the semantics (part of speech and other subtleties) are heavily dependent on the surface realization of the word. Therefore, morphological analysis is inevitable to develop any NLP tool for Indian languages. Unlike English and various foreign languages, most of the Indian languages can be characterized by a rich system of inflections, derivation and compound formation. The numbers of word are derived from the root word by some specific orthographic rule in the Indian languages. A competent morphological analyzer is needed for a machine to deal with the lexicons of these languages.

VII. MORPHOLOGICAL ANALYZER FOR CLASSICAL LANGUAGES

A. Arabic Morphological Analysis and Generation

Kenneth R. Beesley at the Xerox Research centre Europe has been developed an Arabic Morphological Analysis and Generation. Arabic Morphological Analyzer and generator, which was built using Xerox Finite-State Technology. The system also accepts Modern Standard Arabic words and returns morphological analyses and English glosses [10].

B. Hebrew Morphological Analyzer

A Finite-state based morphological analyzer for Hebrew was developed by the by Shlomo Yona. They developed a Morphological Analyzer for un dotted Hebrew words that is based on Finite-state linguistically motivated rules and a broad coverage lexicon. The lexicon contains base forms of words and linguistic attributes that are used by the rules to allow analysis and generation of Hebrew words. The current set of rules comprehensively covers the morphological phenomena that are observable in contemporary Hebrew texts. The analyzer produces output for over 90% of the tokens observed in daily newspapers [4].

C. Greek Morphological Analyzer

Morphological Analyzer of ancient Greek has been developed by David W. Packrd under the Innovative Projects in University Instruction, University of California. The goal was to develop a new textbook and curriculum for teaching ancient Greek to American students. They have prepared statistical summaries of the morphology of each text,, as well as complete concordances organized both according to dictionary lemma and morphological category in this regarding they developed the automated system for morphological analysis [5].

D. Latin parser and translator

The Latin Parser and Translator were developed by Adam McLean to translate form Latin to English. The alternative translations for ambiguous words have been extended and the user can now edit, within the program, the Latin as well as English Translation files [6].

E. Sanskrit Morphological Analyzer

The inflectional Morphology Analyzer for Sanskrit is developed at special Centre for Sanskrit Studies, Jawaharlal Nehru University, New Delhi that identifies and analyzes inflected noun forms and verb-forms in any given sandhi-free text. The system which has been developed as java server RDBMS with Sanskrit data as Unicode text, subsequently, the separate systems of subanta and tinanta will be combined into a single system of sentence analysis with karaka interpretation [7].

F. Chinese Morphological Analyzer

Tseng and Chen developed a Morphological Analyzer for Chinese; their task is to automatically analyze the morphological structures of compounds words. The morphological structures of compound words contain essential information regarding their syntactic and semantic characteristics. [8].

G. Persian Morphological Analysis

A finite-state Morphological Analysis of Persian is developed by Karine Megerdoomian, Department of Linguistics, University of California, USA. The analyzer describes a two-level Morphological Analyzer for Persian using a system based on the Xerox Finite State tools. Persian language presents certain challenges to computational analysis [9].

VIII. RULE BASED APPROACHES IN NATURAL LANGUAGE PROCESSING

This paper we address our successful efforts that involved rule-based approach for Morphological Analyzer for Classical Tamil texts for postpositions.
The rule-based approach has successfully been used in developing many natural language processing systems [10]. The linguistic knowledge acquired for one natural language processing system may be reused to build knowledge required for a similar task in another system. Systems that use rule-based transformations are based on a core of solid linguistic knowledge. The advantage of the rule-based approach over the corpus-based approach is clear for: less-resourced languages, for which large corpora, possibly parallel or bilingual, with representative structures and entities are neither available nor easily affordable, and for morphologically rich languages, which even with the availability of corpora suffer from data sparseness. These have motivated many researchers to fully or partially follow the rule-based approach in developing their natural language processing Analysis and Applications.

IX. RULES FOR POSTPOSITIONS

1. Postpositions

A word that shows the relation of a noun or pronoun to some other word in a sentence. A postposition is similar in function to a preposition, but it follows rather than precedes the object. The suffix list and rules for postpositions are given below

The Classical Tamil Postposition suffix word is listed below

{akam, atu, ayal, aḷavu, aḷavai, aṉ, ā, āṅkaṇ, āṅku, ātu, āyiṭai, āl/āṉ, āṟu, iṭam / iṭaṉ, iṭai, uṭaṉ, uṭai, umpar, uḻi, uḻai, uḷ, uḷḷum, uḷi, ūṅkaṇ, ūṅku, ūru, ūḻ, etirē, ellai, kaṭai, kaṇ, kāl, kāṟum, kīḻ, koṇṭu, cār, ciṟai, ñāṅkaḷ, ñāṉṟu, koṉṟai, talai, tiṟam / tiṟaṉ, niṉṟu, pakkam, paṭi, pāṅkaḷ, pāṭu, piṉ, piṉṟai }

A. Rules for Postpositions

The Morphological Analyzer has three tasks: Training, Rules and tag. The three different level task for Morphological Analyzer The training task trained the validated data into base-level training module using normalization/tokenization. A process of organizing data to tokens from a given corpus is called normalization [12]. Using the normalization theory the special characters k,c,t,p,v end of the any word will be removed.
Ex.

tāḻ irum kūntal eṉ tōḻiyaik kai kaviyāc

(kali. 42:29)

The rules for postpositions are first read the input files then identify the words, if need it will normalized the words. Check the root word dictionary is the database if it is available is will assign the appropriate tag else check the suffix list form the postpositions suffix list remove the suffix and assign as POS, then again check the remaining word in the root words if need apply the rules. The rules procedure for Postpositions is explained as follows

1. Read the input files
2. Indentify the words one by one
3. Normalization / tokenization
4. Check the root word dictionary { if yes assign the appropriate tag }
5. else

ISSN 2348 – 7968

6. Remove the suffix { akam, atu, ayal, aḷavu, aḷavai, aṉ, ā, āṅkaṇ, āṅku, ātu, āyiṭai, āl/āṉ, āṟu, iṭam / iṭaṉ, iṭai, uṭaṉ, uṭai, umpar, uḷi, uḻai, uḷ, uḷḷum, uḷi, ūṅkaṇ, ūṅku, ūru, ūḻ, etirē, ellai, kaṭai, kaṇ, kāl, kāṟum, kīḻ, koṇṭu, cār, ciṟai, ñāṅkaḷ, ñāṉṟu, koṉṟai, talai, tiṟam / tiṟaṉ, niṉṟu, pakkam, paṭi, pāṅkaḷ, pāṭu, piṉ, piṉṟai }
7. Assign the suffix tag { POS }
8. Check the root word dictionary
9. If yes { assign the appropriate tag }
10. If no function call { add ‘u’ rule / add ‘doubling’ rule / add ‘sandthi’ rule }
11. Check the word in the root word dictionary
12. Assign the tag { NN } only noun
13. Stop

B. Function Rules

The following function rules are executed when needed for postposition rules.

a. Add ‘u’ rule

The add ‘u’ rule executes when the word is not available in root dictionary and end of the character is consonant.

Function add_u( )

{

1. After remove the suffix
2. Check the last character
3. if the character ‘consonants’
4. Add the end of the suffix ‘u’
5. Check the root word dictionary
   { Assign the appropriate tag }
   { go to 7 }
6. Else check the root word dictionary
   { Assign the appropriate tag }
7. Stop

}

b. doubling rule

The doubling rule is executes when the word not available in the in the root dictionary and the two characters are same as listed.

Function double( )

{

1. After remove the suffix
2. Check the last two characters { if the ‘ṇṇ, mm, ll, yy, ṉṉ’ }
3. Remove a single character {Check the dictionary and assign the Tag }
4. Else if the last two characters { ṭṭ, ṟṟ }
5. Remove one character and assign the tag for remaining character
6. Add ‘u’ in the end of the remaining word { go to add ‘u’ rule }
7. Check the dictionary and assign the appropriate tag
8. Stop

}

c. Sandthi rule

The sandhi rule is executed when the word is not available in the root word dictionary and the last character is same as listed.

Function sandhi( )

{

1. Check for root word dictionary { if yes assign the appropriate tag }
2. If no check the suffix { Remove the suffix and assign the Tag }
3. Check the remaining word
4. If the word end with ‘y’ { remove the suffix and assign the Tag }
5. Check the dictionary and assign the appropriate tag
6. Stop

}

d. Examples

1. ātu

puṟavil taṉātu ceṅkatir celvaṉ kataṟutaliṉ (naṟ. 164.1-2)

2. āl/āṉ

oḷ oḷi maṇip poṟiyāṉ mañcai (pari 18:7)

naṉṉār naṉmoḻi kēṭṭaṉam ataṉāl

3. piṉ

teḷ kaṭal cērppaṉaik kaṇṭa piṉṉē. (kuṟu. 306:6)

4. ūḻ

ellaiyum iravum tuyil tuṟantu pal ūḻ (kali. 123:16)

ISSN 2348 – 7968

X. PROCEDURE

The Morphological Analyzer for rule-based approach contains a set of rules and a dictionary that contains root words and morphemes. The root word dictionary database was developed from Classical Tamil texts which is stored in the form of XML database. The major eight grammatical categories are Noun, Pronoun, Particle, Verb, Case Marker, Clitics, Conjunction, Demonstrative, Post Position. This database is used for identify the root words of the Classical Tamil corpus. The schema of the database is { dictionary_ID, root_WordCategory, word }[11].

a. Procedure

The procedure for the Morphological Analyzer is the inputted Classical Tamil corpus is assigned by the word by word. A word is first check from the root word dictionary if the word is available in the dictionary is assigns its grammatical category else it go for the rules the, the rules for postpositions the rules is first take list of postpositions and the procedure is start from reverse characters. First remove the postpositions and assign it as POS the remaining word is check into the dictionary if it available assign its grammatical category otherwise word is go for add 'u' rule / doubling rule / sandhi rule. The following fig 1.1 explains the procedure for the Morphological Analyzer

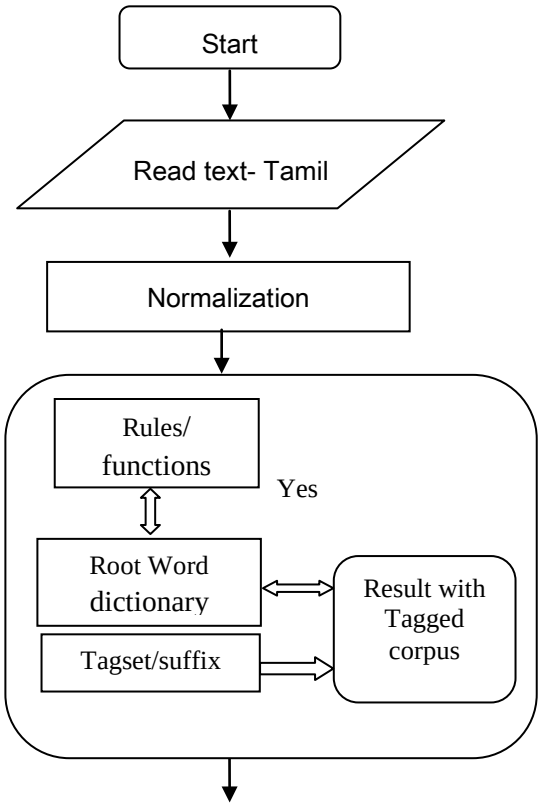

Fig 1.1 Procedure for Morphological Analyzer

The add 'u' rules acts when the last character of the remaining word is consonant add 'u' and check the root word dictionary assign its grammatical categories. The doubling rules acts when the last two characters is 'ṇṇ, mm, ll, yy, ṉṉ' the rule will remove the one character then check the root word dictionary assign its grammatical categories. The sandhi rules acts when the last character is 'y' the last character will remove and check the root word dictionary and assign its grammatical categories.

XI. ANALYSIS

The Morphological Analysis identifies root and suffixes of a word and assigns its grammatical categories. Some of the approaches are used for Morphological Analyzers. The rule-based approaches are produces the more accuracy of results. The rule-based approach used for morphological analysis which are based on a set of rules and dictionary that contains root words and morphemes. In rule-based approach, a particular word is given as an input to the morphological analyzer and if that corresponding morpheme or root word is missing in the dictionary then the rule-based system fails. Here each rule depended on the previous rule. So if one rule fails, it affects the entire rule that follows. In the course of testing of the rule, certain inconsistencies and lapses in recognizing certain word, The above rules is applied in the Morphological Analyzer for Classical Tamil texts is produces the 72 percentage of accuracy for Classical Tamil texts.

References

[1] Anandan. P, RanjaniParthasarathy, Geetha T.V.2002. Morphological Analyzer for Tamil, ICON 2002,RCILTS-Tamil, Anna University, India

[2] www.en.wikepedia.org

[3] Andronov, M. 1969. The Standard Grammar of Modern and Classical Tamil. Madras: New Century Book House, Pvt. Ltd.

ISSN 2348 – 7968

[4] Shlomo Yona, November, 2004, A Finite-state based morphological analyzer for Hebrew, Faculty of Social Science Department of Computer Science, University of Haifa.

[5] David W. Packrd COMPUTER-ASSISTED MORPHOLOGICAL ANALYSIS OF ANCIENT GREEK at Innovative Projects in University Instruction, University of California.

[6] http://www.logosconjugator.org/newverb/verba_dba.verba_main.create_page?lang=en

[7] Girish Nath Jha, Muktanand Agrawal, Subash, Sudhir K. Mishra, Diwakar Mani, Diwakar Mishra, Manji Bhadra, Surjit K. Singh "Inflectional Morphology Analyzer for Sanskrit" at Special Centre for Sanskrit Studies, Jawaharlal Nehru University, New Delhi

[8] Huihsin Tseng and Ken-Jiann Chen., 2002 Design of Chinese Morphological Analyzer. In procedings of First SINGHAN Workshop

[9] San Diego, Linguistics Department, University of California, USA And Karine Megerdoomian, Inxight Software "Finite-State Morphological Analysis of Persian"

[10] Khaled Shaalan, June 2010, "Rule-based Approach in Arabic Natural Language Processing" in International Journal on Information and Communication Technologies, Vol. 3, No. 3

[11] Ankita Agarwal, Pramila, Shashi Pal Singh, Ajai Kumar, Hemant Darbari, March-2014 " Morphological Analyser for Hindi – A Rule Based Implementation " International Journal of Advanced Computer Research (ISSN (print): 2249-7277 ISSN (online): 2277-7970) Volume-4 Number-1 Issue-14.

[12] M. Mohamed Yoonus and Samar Sinha, September 2011 "A Hybrid POS Tagger for Indian Languages" an e journal of Languages in India (www.languageinindia.com), Volume 11: 9