International Journal of Latest Trends in Engineering and Technology
Special Issue SACAIM 2016, pp. 19-24
e-ISSN:2278-621X

MALAYALAM MORPHOLOGICAL ANALYZER USING HFST: AN APPROACH

Ajusha P V[1], Babu Anto P[2]

Abstract- Morphological Analysis of a language is crucial in any Natural language Processing tasks. The grammar of a given language can be identified using a Morphological Analyzer . For morphologically rich and agglutinative languages like Malayalam identification of constituents of the words using a Morphological Analyzer, is a tedious task. Different techniques are already developed for this task. This paper tries to describe an approach to handle the Malayalam Morphological analyzer using the HFST. The lexicon compilation tool, HFST-LEXC and the two-level grammar rules compilation tool, HFST-TWOLC are also discussed.

Keywords – Morphological Analysis, Malayalam, HFST,

I. INTRODUCTION

Natural Language Processing (NLP) is aims to develop automated tools for language processing[1]. In NLP Morphological Analysis is used to identify the morpheme and suffixes of words in a language and individual words are analyzed into their components. The morphemes are the stem, which is the meaning-bearing word and affixes add extra meaning to the stem.

Dravidian Language like Malayalam is morphologically rich and agglutinative in nature. So the development of an efficient Morphological Analyzer for Malayalam is a tedious task. The inflections, multiple suffixes, and word compounding are the major barriers for Malayalam Morphological Analysis. A number of attempts to build a morphological analyzer for Malayalam were carried out in recent years. But a full-fledged Morphological analyzer with high performance is still a difficult challenge. Transducers can be used for phonological processes and relate various linguistic abstraction levels, using tools like TWOLC introduced by Koskenniemi and Karttunen[2].

[1] *School of Information Science and Technology Kannur University*
[2] *School of Information Science and Technology Kannur University*

II. RELATED WORKS

Several approaches have already experimented in the development of Malayalam morphological analyzer. Root driven method and Affix stripping methods are the most commonly used methods for Malayalam Morphological Analysis. Suffix stripping method for the root word identification is done using the finite state transducers[3]. The corpus-based approach is another way of doing morphological analysis in Malayalam[4] in which the results depends on the content of corpus used. For inflectionally rich languages paradigm approach [5][6] can give results depends on the contents of the paradigm. Root word identification through rule-based approach[7][8] deals with the identification of root words and remove inflections.

III. MALAYALAM MORPHOLOGY

Morphology for every language is different. Malayalam words can occur in its root form, inflected form, derived form, compound form and in reduplicated form[7]. The Morphological Analyzer separates the free morphemes called root and suffix morphemes or bound morphemes in every lexical item. In the term *pEnakaL, pEna*is the root word and *kaL* is the suffix. The main grammatical categories of Malayalam are noun, pronoun, verb, postpositions adverb, adjectives etc.

A. Orthographic (Sandhi) Rules

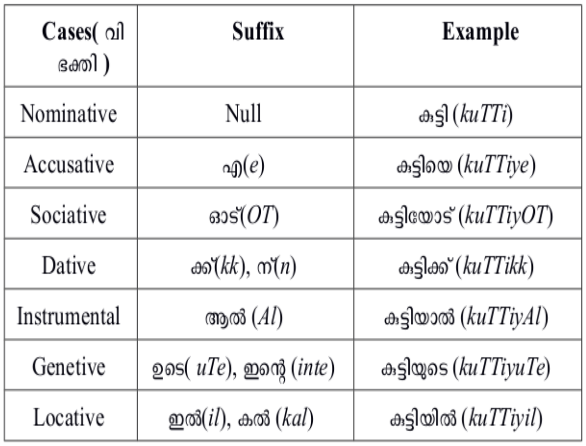
The phonological changes that occur at morpheme boundaries while combining words are called Sandhi. To split the compound word sandhi rules can be used. There is both internal and external sandhi. The sandhi exists between a root or a stem with asuffix or a morpheme is an Internal sandhi. External sandhi is between words.Two or more words join to form a single string ofconjoined words[9].The Malayalam sandhi rules are Elision(*loopa*), Augmentation(*aagama*), Reduplication(*dvita*), and Substitution(*aadeesa*).

B. Noun Morphology

In Malayalam, nouns can be inflected due to plural markers and case markers. Plural markers are grammatical numbers. If the noun is human then the plural marker *mAr* and for non-human the plural marker *kAl* is used generally. There can be some exceptions to this rule for words like *peN*for which the plural is *peNungaL*.

The inflections due to case markers relate the nouns in a sentence. There are seven case markers in Malayalam. They are shown in Table 1.

Table-1 Cases in Malayalam



Cases(വി ഭക്തി)	Suffix	Example
Nominative	Null	കുട്ടി (*kuTTi*)
Accusative	എ(*e*)	കുട്ടിയെ (*kuTTiye*)
Sociative	ഓട്(*OT*)	കുട്ടിയോട് (*kuTTiyOT*)
Dative	ക്ക്(*kk*), ന്(*n*)	കുട്ടിക്ക് (*kuTTikk*)
Instrumental	ആൽ (*Al*)	കുട്ടിയാൽ (*kuTTiyAl*)
Genetive	ഉടെ(*uTe*), ഇന്റെ (*inte*)	കുട്ടിയുടെ (*kuTTiyuTe*)
Locative	ഇൽ(*il*), കൽ (*kal*)	കുട്ടിയിൽ (*kuTTiyil*)

Postpositions like *aaya*, *aayi*, *vENTi*and *ninnu* are acts as the secondary case suffixes.

C. Verb Morphology

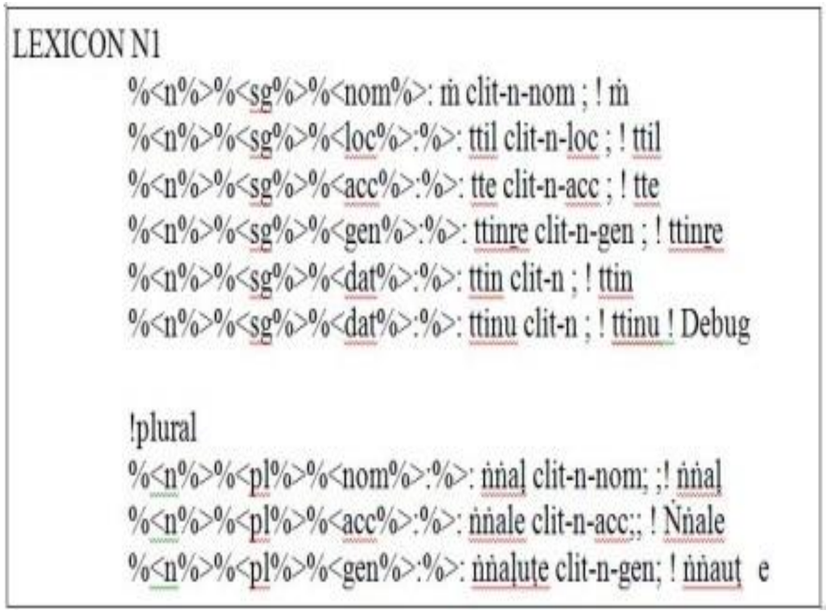
Verbs denote action. Malayalam verbs get modified due to mood, tense, aspect negation and voice. There are three base forms for Malayalamverbs , they are Intransitive, Transitive, and Causative. Tense can be classified into past(*bhUtaM*), present(*vartamAnam*) and future(*bhAvi*). Aspect as perfective, imperfective, progressive. Mood form generated are classified as denotative ,interrogative, purposive, imperative, conditional, optative, and potential. The two types of voices are active voice and passive voice[7]. The suffix *illa*added for marking negation.

IV.HFST

In natural language processing, finite-state string transducer methods have been found useful for solving a number of practical problems ranging from language identification via morphological processing and generation to part-of-speech tagging and named-entity recognition, as long as the problems lend themselves to a formulation based on matching and transforming local context[10]. For morphological processing of agglutinative languages, finite-state string transducer methods were found to be useful[10]. The main advantage of HFST is that it provides an interface to an increasing number of software libraries for processing finite-state transducers. There two main files in the morphological transducer in HFST-LEXC and HFST-TWOL files. The HFST-LEXC defines morphotactics which gives the information about how morphemes in the are joined together in a word. The HFST-TWOL(two-level rules) describemorphophonology, i.e. what changes happen when these morphemes are joined together.

A.HFST-LEXC

LEXC is the lexicon compiler create a finite-state transducer of a lexicon by reading the morpheme sets and their morphotactic combinations and are called as lexicon transducers. In LEXC, morphemes are grouped into named sets called sub-lexicons. Each entry of a sub-lexicon is a pair of finite possibly empty strings separated by ’:’ and associated with the name of a sub-lexicon called a continuation class[11]. An example of the lexicon for non-human noun like മരം(maram), which can have seven inflections due to cases i.e., nominative, dative, instrumental, locative, accusative, and sociative and can also be classified on the basis of number ,singular, and plurals, is shown below.



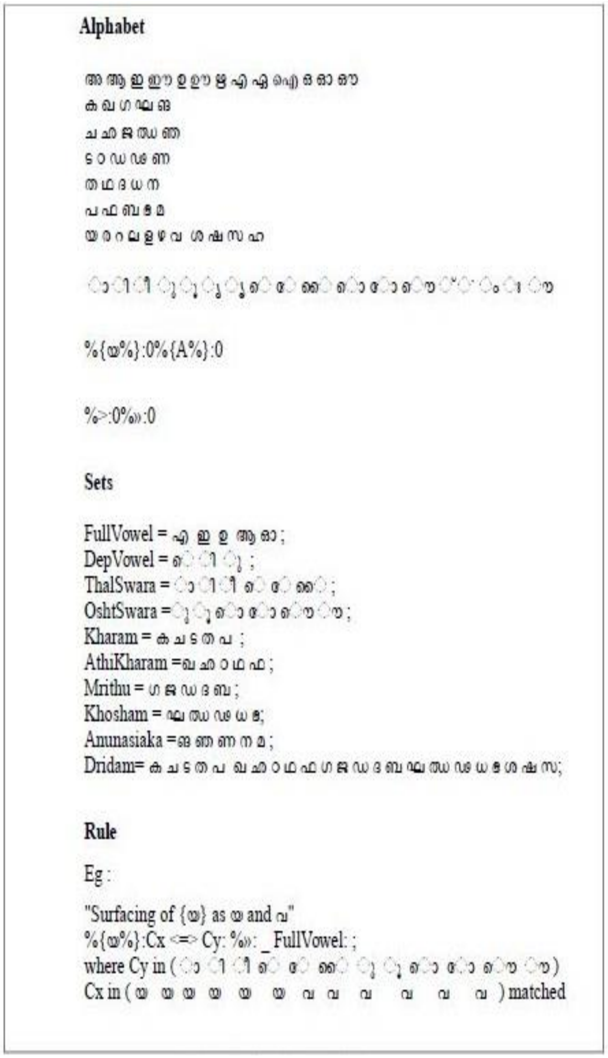
```
LEXICON N1
          %<n%>%<sg%>%<nom%>: ṁ clit-n-nom ; ! ṁ
          %<n%>%<sg%>%<loc%>:%>: ttil clit-n-loc ; ! ttil
          %<n%>%<sg%>%<acc%>:%>: tte clit-n-acc ; ! tte
          %<n%>%<sg%>%<gen%>:%>: ttinṟe clit-n-gen ; ! ttinṟe
          %<n%>%<sg%>%<dat%>:%>: ttin clit-n ; ! ttin
          %<n%>%<sg%>%<dat%>:%>: ttinu clit-n ; ! ttinu ! Debug

          !plural
          %<n%>%<pl%>%<nom%>:%>: ṅṅaḷ clit-n-nom; ;! ṅṅaḷ
          %<n%>%<pl%>%<acc%>:%>: ṅṅaḷe clit-n-acc;; ! Ṅṅaḷe
          %<n%>%<pl%>%<gen%>:%>: ṅṅaḷuṭe clit-n-gen; ! ṅṅauṭ   e
```

Figure-1 Sample Lexicon

Paradigms for nouns, verbs, proper nouns, pronouns, numerals, adjectives, and adverbs can be added according to their morphological behaviors.

B. HFST-TWOL

Two-level rules for HFST-TWOL are parallel constraints on symbol-pair strings governing the realizations of lexical word-forms as corresponding surface-strings[11] . They used for modeling the phonology of languages. HFST-TWOLC is the two-level rule compiler which compiles the grammars of two-level rules into a set of finite-state transducers. It takes the surface forms produced by LEXC and apply rules to them to alter them into surface forms. The alphabets , rule variables, sets are written to form the rules. The types of rules can be categorized to Phonologically conditioned deletion, Morphologically conditioned deletion, Phonologically conditioned symbol change, Morphologically conditioned symbol change, Phonologically conditioned insertion, and Morphologically conditioned insertion. The Malayalam sandhi rules are used for handling the agglutination. An example of a simplified grammar for the surface realization is shown below.



```
Alphabet

അ ആ ഇ ഈ ഉ ഊ ഋ എ ഏ ഐ ഒ ഓ ഔ
ക ഖ ഗ ഘ ങ
ച ഛ ജ ഝ ഞ
ട ഠ ഡ ഢ ണ
ത ഥ ദ ധ ന
പ ഫ ബ ഭ മ
യ ര റ ല ള ഴ വ ശ ഷ സ ഹ

◌ാ ◌ി ◌ീ ◌ു ◌ൂ ◌ൃ ◌ൄ ◌െ ◌േ ◌ൈ ◌ൊ ◌ോ ◌ൌ ◌് ◌ഁ ◌ം ◌ഃ ◌ൗ

%{യ%}:0%{A%}:0

%>:0%»:0

Sets

FullVowel = എ ഇ ഉ ആ ഓ ;
DepVowel = ◌െ ◌ി ◌ു ;
ThalSwara = ◌ാ ◌ി ◌ീ ◌െ ◌േ ◌ൈ ;
OshtSwara = ◌ു ◌ൂ ◌ൊ ◌ോ ◌ൌ ◌ൗ ;
Kharam = ക ച ട ത പ ;
AthiKharam =ഖ ഛ ഠ ഥ ഫ ;
Mrithu = ഗ ജ ഡ ദ ബ ;
Khosham = ഘ ഝ ഢ ധ ഭ;
Anunasiaka =ങ ഞ ണ ന മ ;
Dridam= ക ച ട ത പ  ഖ ഛ ഠ ഥ ഫ ഗ ജ ഡ ദ ബ ഘ ഝ ഢ ധ ഭ ശ ഷ സ;

Rule

Eg :

"Surfacing of {യ} as യ and വ"
%{യ%}:Cx <=> Cy: %»: _FullVowel: ;
where Cy in ( ◌ാ ◌ി ◌ീ ◌െ ◌േ ◌ൈ ◌ു ◌ൂ ◌ൊ ◌ോ ◌ൌ ◌ൗ )
Cx in ( യ  യ യ  യ  യ  യ  വ വ  വ  വ  വ  വ ) matched
```

Figure-2. Sample TWOL Grammar

These rules are compiled and in the rule-conflicts are resolved . A program HFST-COMPOSE-INTERSECT is used to combine the results of HFST-LEXC and HFST-TWOL.

V.CONCLUSION

This paper discusses an approach towards the building of a morphological analyzer for the Malayalam language. For an agglutinative and morphologically rich language like Malayalam, the development of NLP tools is a difficult task. HFST which uses the finite-state methods is one of the ways to implement the Malayalam Morphological analyzer. Since the inflections in Malayalam are marked using suffixes the guessing of the Morphology using a finite-state transducer is found to be effective.

REFERENCES

[1] S. Tanveer , U S Tiwary, “Natural language processing and information retrieval”, Oxford University press.

[2] K. Koskenniemi, “Two-Level Morphology: A General Computational Model for Word-Form Recognition and Production”, Proceedings of 10th International Conference on Computational linguistics, Association for computational linguistics, 1984.

[3] R. R. Rajeev, N. Rajendran, and S. Elizabeth, “A Suffix Stripping Based Morph Analyses For Malayalam Language”, Science Congress, 2007.

[4] V. P. Abeera, S. Aparna, R. U. Rekha, M. Anand Kumar, V. Dhanalakshmi, and K. P. Soman, “Morphological Analyzer for Malayalam Using Machine Learning”, in ICDEM10 Proceedings of the Second international conference on Data Engineering and Management, pp. 252-254, 2012.

[5] J. V. Nimal, B.Narsheedha, “Malayalam Noun and Verb Morphological Analyzer:A Simple Approach”, International Journal of Software & Hardware Research in Engineering, Vol. 2 ,August 2014.

[6] P. M. Vinod ,V. Jayan, and V. K. Bhadran, “Implementation of Malayalam Morphological Analyzer Based on Hybrid Approach”, Proceedings of 24th conference on Computational linguistics and Speech processing, 2012.

[7] S. Meera, M. Wilscy, and S.A. Shanavas, “A Rule Based Approach for Root Word Identification in Malayalam Language”, International Journal of Computer Science & Information Technology (IJCSIT), Vol. 4, No. 3, June 2012.

[8] M. I. Sumam, S. D. Peter, “A Morphological processor for Malayalam Language”, South Asia Research Journal, Vo1. 27(2), h, SAGE Publications U.K, 2007.

[9] V.V. Devadath, J.K. Litton, M. S. Dipti and V. Vasudeva, “A Sandhi Splitter for Malayalam” , in Proceedings of ICON, 2014.

[10] K. Linden, E. Axelson , S. Drobac, S. Hardwick, J. Kuokkala, J. Niemi, T. Pirinen and Silfverberg "HFST—A System for Creating NLP Tools.", International Workshop on Systems and Frameworks for Computational Morphology, Springer Berlin Heidelberg, 2013.

[11] K. Linden, M. Silfverberg, , T. Pirinen, “HFST Tools for Morphology—An Efficient Open Source Package for Construction of Morphological Analyzers”, In*:* Mahlow, Piotrowski (eds.) , pp. 28–47,2009.

[12] K. Lindén, A. Erik, H. Sam, A. P. Tommi, and S. Miikka, "HFST-Framework for Compiling and Applying Morphologies." International Workshop on Systems and Frameworks for Computational Morphology, Springer Berlin Heidelberg, 2011.