Application of Finite State Methods in Malayalam Text Analysis

Girija V. R.
PhD Student
Department of Computer Science
Dravidian University, Kuppam, A. P, India

T. Anuradha
Associate Professor
Department of Computer Science
Dravidian University, Kuppam, A. P, India

ABSTRACT

This paper discusses the design of a morphological analyzer for Malayalam that can be used for text analysis using Finite State Models. Finite state model that recognizes and strips the morphemes in a string of text are dealt with here.

General Terms

Text analysis, Text segmentation, finite state automaton, Finite State Transducer, Morphological analysis.

Keywords

Malayalam morphology, Lexicon, Suffix Extraction, Morphophonemic rules.

1. INTRODUCTION

Text analysis is the process of extracting useful information and knowledge from unstructured or structured text. Natural Language Processing is used to discover relevant information in text by transforming text into data that can be used for further analysis. For analyzing texts, it should be segmented into sentences and then words (tokenization) for further processing. Then these words are segmented into morphemes and their grammatical categories are identified. This process is called morphological analysis which is the primary step for Text Analysis. A morphological analyzer recognizes and strips the morphemes in a string of text. This task involves identifying morphophonemic changes that occur in between the morphemes. Malayalam being a highly agglutinative language wherein the morphology mainly involves concatenation of suffixes. Finite State Method (FSM) is an appropriate computational model for the morphological analysis of Malayalam since its morphology is concatenative in nature.

FSM's are widely used in the morphological analysis of wide variety of agglutinative languages. In [1], morphological analysis of kazakh language was implemented by two level morphology and Foma finite state tools. The finite state approach was applied to analyze morphology of the Bishnupriya Manipuri language[2]. A Morphology engine for Telugu verbs was developed based on finite state techniques[3]. The design of finite state machine in the development of Malayalam spell checker was discussed in [4]. An algorithm has been developed for splitting the Malayalam compound words[5]. The splitter made use of lexicons, which were created as tries. Here the compound words were scanned from left to right for splitting. But in Malayalam, suffixes are limited when compared to the stem and also the type of the suffix will determine the category of stem to an extent, it would be better to approach a word from right to left for extracting suffixes. The objective of this paper is to propose a method for morphological analysis of Malayalam texts using FSM based suffix stripping. In the proposed method, stripping of morphemes are carried out from right to left which makes it computationally economic.

The Finite State Transducers (FST) are used to translate an input string into an output string. A morphological analyzer using FST segments a string of text into morphemes and lexemes. For example, if the string kiLikaLe (birds+accusative) is given as input, the output is kiLi (N) +PL+ACC. The Sandhi rules (morphophonemic rules) in Malayalam can also be handled by transducers as in the language phonemic changes on concatenation occurs to the phonemes in the morpheme boundaries only. For eg: marangal (trees) → maram+kal, pasukkal (cows) -> pasu+kal. The rules of these morphophonemic changes can be modeled using Finite State Transducer as the change depends on the last phonemes of the preceding and the first phonemes of the following morpheme.

2. TEXT SEGMENTATION

Text segmentation is the process of dividing written text into linguistically meaningful units such as sentences and words [6]. It is an essential part of Natural language processing system. The words and sentences identified at this stage are passed to further processing such as morphological analysis, POS tagging, parsing etc.

Sentence segmentation is the process of dividing a string of written language into its component sentences. This involves identifying sentence boundaries. In Malayalam, punctuation marks such as period (.), question mark (?) and exclamation mark (!) indicates sentence boundary. The sentence boundary disambiguation can be handled by checking the preceding token of a period character in a list of abbreviations. Word segmentation breaks up the sequence of characters in a sentence by locating the word boundaries. As Malayalam is a space delimited language, the space is a good approximation of a word delimiter.

3. FINITE STATE METHOD

It is a computational model that has an initial state and can be in exactly one of a finite number of states at any given time. The Finite State Model can change from one state to another in response to some external inputs, which is called a transition. The transition property reveals that finite state models such as finite state automata and finite state transducers can be effectively used in a wide range of domains including natural language processing, optical character recognition, speech processing, data compression etc. They are formally defined as follows.

A finite state automaton (FSA) is a 5-tuple (Σ, Q, i, F, E) where Σ is a finite set of alphabets, Q is a finite set of states, $i \in Q$ is the initial state, $F \subseteq Q$ is the set of final states and $E \subseteq$

$Q \text{ x } (\Sigma \cup \{\varepsilon\}) \text{ x } Q$ is the set of edges [7]. Fig. 1 represents an FSA that recognizes well, tell, wall, tall.

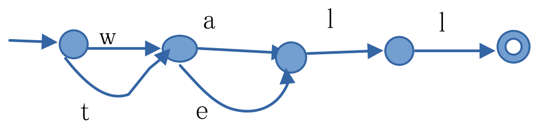


Fig. 1: FSA that recognizes well, tell, wall, tall

A Finite State Transducer (FST) is an FSA, in which each arc is labeled by a pair of symbols. A FST is a 6 –tuple (Σ_1, $\Sigma_{2,}$ Q, i, F, E) such that Σ_1 is a finite set of input alphabet, Σ_2 is a finite set of output alphabet, Q is the finite set of states, i $\in$ Q is the initial state, F $\subseteq$ Q is the set of final states, E $\subseteq$ Q x Σ_1x Σ_2 x Q is the set of edges [7]. Fig.2 represents an FST that maps look, looks, looked to look.

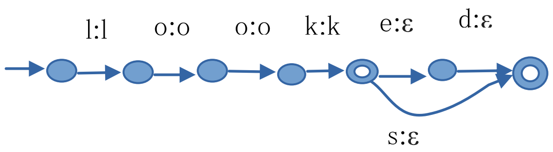


Fig. 2: FST that maps look, looks, looked to look

The various stages of computational morphological analyzer can be effectively modeled using FSA and FST. The lexicons can be implemented using finite state automata. Thus they outperform indexation in terms of speed and memory space consumption. The transducers can be considered as a representation for mapping from strings to strings. Thus FST can be used to arrive at the lexical level from the surface level. The morpheme ordering can be dispensed with using FST that relate surface and lexical levels directly. Morphophonemic changes may occur during stripping suffixes from a word. These context sensitive rewriting rules can be implemented using FST. Finite state transducers are used for phonological and morphological analysis and morpheme sequencing in natural language processing. The closure and algorithmic properties of finite state models reveal that it is very powerful and flexible for modeling natural languages [8]. The class of epsilon free transducers is closed under intersection and has therefore been used extensively in morphology and phonology.

4. MORPHOLOGICAL ANALYSIS

Morphological analysis is the primary step of any natural language processing task. It is the segmentation of words into their component morphemes and the assignment of grammatical morphemes to grammatical categories and lexical morphemes to lexemes. For example kiLikaL (birds) could be analyzed as kiLi (Noun Stem, NS) + Plural (PL). The main challenge of this task is choosing an appropriate model which considers the complexity of the morphology of the language being considered.

5. MALAYALAM MORPHOLOGY

Malayalam language is one among the 22 scheduled languages in India. It belongs to the Dravidian family of languages and designated as a classical language in India in 2013. Malayalam is highly agglutinative and morphologically rich language [9]. In the language, the words are formed by concatenation of suffixes. For example, noun can take suffixes such as plural marker, case marker, post positions, clitics etc. Verb can take tense, aspect and mood. When morphemes combine to form words, morphophonemic alterations may happen.

6. METHODOLOGY

Morphology of a language deals with the stems and suffixes and the rules by which they are combined to form words. Computational morphological analysis of Malayalam text will need lexicons (lists of noun and verb stems), lists of suffixes and list of morphophonemic rules. The morphology and lexicons are handled using finite state methods. Fig.3 represents the block diagram for morphological analysis.

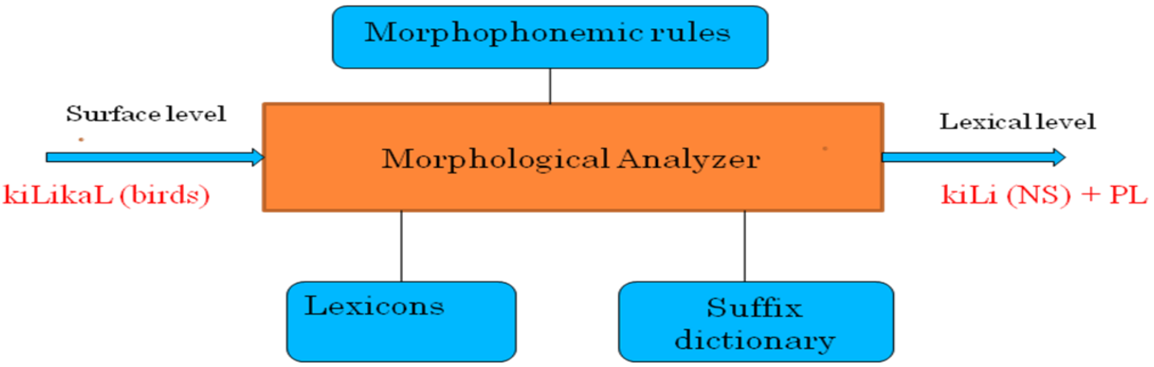
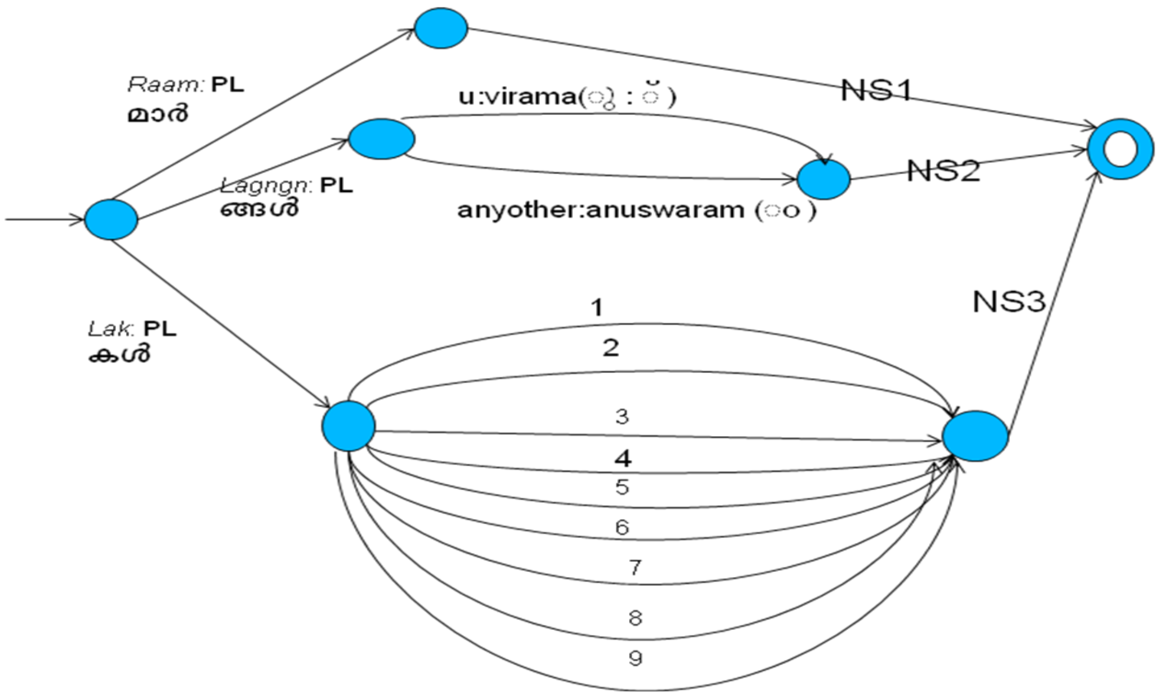


Fig. 3: Block diagram for morphological analysis

6.1 Suffix Extraction

As Malayalam is an agglutinative language, most of the words are formed by adding one or more suffixes to the stem. Morphological analysis needs to extract suffixes from the word and assign grammatical categories to these suffixes. Morpheme ordering or sequencing can be modeled with finite state transducers. Thus it can be used to generate lexical level from the surface level. In Malayalam, as the suffixes are limited when compared to the stem, it would be better to approach a word from right to left for extracting suffixes [10]. Also grammatical category of suffixes will distinguish the category of stem to an extent.

For example Noun inflections can be of the form NS ± PL ± CS ± CL ± PP ± CL ± PP ± CL where NS :noun stem PL :plural suffix, CS :case suffix, CL :clitics, PP :postposition. Fig. 5 represents an FST for extracting plural suffixes from nouns



NS : Noun Stem PL : Plural

1 nu : n (നു : ൻ)
2 Nu : N (ണു : ൺ)
3 lu : l (ലു : ൽ)
4 Ru : R (റു : ർ)
5 ru : R (രു : ർ)
6 Lu : L ((ളു : ൾ)
7 other u : virama (other ു : ്)
8 k : ε (ക് : ε)
9 any other : ε

Fig. 5: FST for extracting plural suffixes from nouns

Some examples of the output of the above FST for different categories of nouns are:

NS1: ammamaaR→amma (NS) + PL

അമ്മമാർ→ അമ്മ (NS)+ PL

NS2: marangaL→maram (NS)+PL

മരങ്ങൾ→ മരം (NS)+ PL

KunjungaL→kunju (NS)+PL

കുഞ്ഞുങ്ങൾ→ കുഞ്ഞ് (NS)+ PL

NS3: pasukkaL→pasu (NS)+PL

പശുക്കൾ → പശു (NS)+ PL

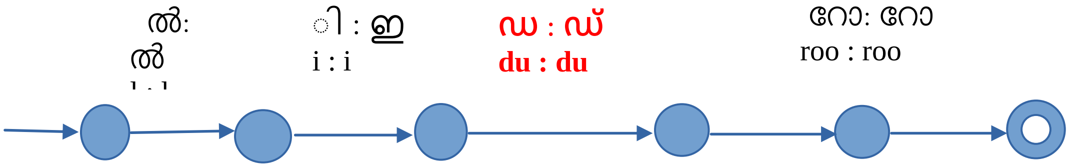
vaLakaL→vaLa (NS)+PL

വളകൾ → വള (NS) + PL

6.2 Morphophonemic rules

In Malayalam, Words of different grammatical categories are combined to form a single word. A new word may be formed by combining noun and noun, noun and adjective, verb and noun, adverb and verb, adjective and noun. Morphological analyzer needs to split these words into morphemes. Morphophonemic rules should be considered while splitting the compound words. Morphophonemic alterations must be taken in to account in suffix extraction stage also. Most morphophonemic alterations depend on one or a few neighboring phonemes. These context sensitive rewriting rules can be modeled as finite state transducers.When two words or characters join, morphophonemic changes may occur at the point of joining which is called Sandhi. Sandhi can be classified into Elision (Lopa Sandhi), Augmentation (Agama Sandhi), Reduplication (Ditva Sandhi) ,Substitution(Adesha Sandhi) [11]. Sandhi rules (Morphophonemic rules) are formed by considering the end phoneme of the first morpheme and start phoneme of the second morpheme. Morphological analyzer takes essence of these rules and handles morphophonemic changes occurred in the morpheme boundary.

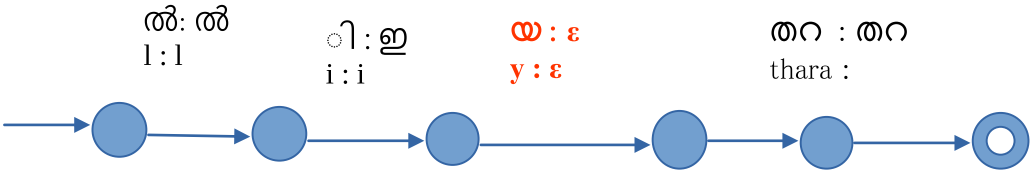
When a vowel at the beginning of a morpheme comes after consonant, schwa is added to the consonant (Elision). Fig. 6 shows the handling of elision using FST When a vowel at the beginning of a morpheme comes after ya / va (യ / വ), then ya /va (യ / വ) is deleted (Augmentation). Fig. 7 shows the handling of augmentation using FST. When the consonant at the beginning of a morpheme comes after same consonant along with chandrakkala (virama ്), then consonant along with chandrakkala is deleted (reduplication). Fig. 8 shows the handling of reduplication using FST. In Substitution, one letter is substituted by another during concatenation. Fig. 9 shows the handling of substitution using FST.



roodil -> roodu + il

റോഡിൽ -> റോഡ് + ഇൽ

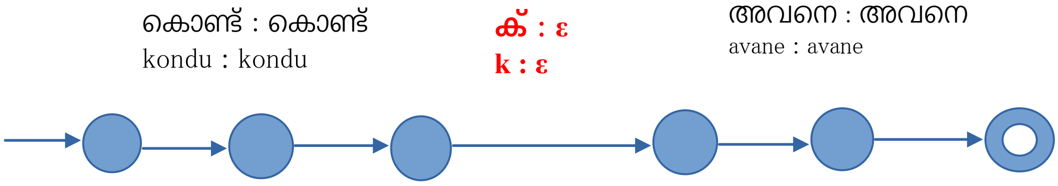
Fig. 6: Handling of elision using FST



tharayil -> thara + il

തറയിൽ -> തറ+ഇൽ

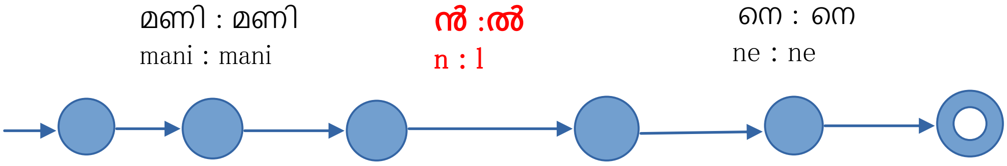
Fig. 7: Handling of augmentation using FST



avanekkondu -> avane + kondu

അവനെക്കൊണ്ട് -> അവനെ + കൊണ്ട്

Fig. 8: Handling of reduplication using FST



nenmani -> nel + mani

നെന്മണി -> നെൽ + മണി

Fig. 9: Handling of substitution using FST

6.3 Lexicons

A computational morphological analyzer requires lists of stems (noun and verb) and lists of suffixes (case suffix, postpositions, past tense markers etc.). As there are a large number of stems which are homographic in nature, it is better to store stem list using finite state technology. Finite state technology will reduce the memory space consumption. It also offers very fast lookup because the recognition doesn't depend on the size of the list but only on the input strings considered [5]. Thus optimal time and space efficiency can be attained by its indexation. Fig. 4. represents a part of lexicon of noun stems using FSA.

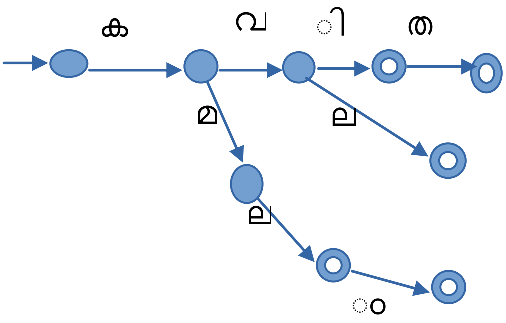


Fig. 4: Representation of noun stems using FSA

7. CONCLUSION

This paper presents a model that recognizes and strips the morphemes in a string of text. The closure and algorithmic properties of finite state models make them a very powerful and flexible for modeling natural languages. Malayalam lexicons can be modeled using finite state automata. Thus optimal time and space efficiency can be attained. Finite state transducers are used for phonological and morphological analysis and morpheme sequencing in Malayalam language processing. Thus Finite State Method provides appropriate and linguistically motivated models for implementing Malayalam morphological analysis.

Though designed for text analysis, this method can be used for other areas of applications in Natural language processing. Since morphology is closely related to syntax and semantics, incorporating syntactic and semantic rules will improve the accuracy of the proposed method.

8. REFERENCES

[1] Kessikbayeva, Gulshat, and Ilyas Cicekli. "A rule based morphological analyzer and a morphological disambiguator for kazakh language." Linguistics and Literature Studies 4.1 (2016): 96-104.

[2] Kalita, Nayan Jyoti, Navanath Saharia, and Smriti Kumar Sinha. "Morphological Analysis of the Bishnupriya Manipuri Language using Finite State Transducers." International Conference on Intelligent Text Processing and Computational Linguistics. Springer Berlin Heidelberg, 2014

[3] Dokkara, Sasi Raja Sekhar, Suresh Varma Penumathsa, and Somayajulu G. Sripada. "Verb Morphological Generator for Telugu." *Indian Journal of Science and Technology* 10.13 (2017).

[4] Manohar, Nimtha, et al. "Spellchecker for Malayalam using finite state transition models." *Intelligent Computational Systems (RAICS), 2015 IEEE Recent Advances in*. IEEE, 2015.

[5] Nair, Latha R., and S. David Peter. "Development of a rule based learning system for splitting compound words in malayalam language." Recent Advances in Intelligent Computational Systems (RAICS), 2011 IEEE.

[6] Dale, Robert, Hermann Moisl, and Harold Somers, eds. Handbook of natural language processing. CRC Press, 2000.

[7] Roche, Emmanuel, and Yves Schabes. Finite-state language processing MIT press, 1997

[8] Jurafsky, Dan. Speech & language processing. Pearson Education India, 2000.

[9] Asher, Ronald E., and T. C. Kumari. *Malayalam*. Psychology Press, 1997.

[10] Valath, Nimal J., and Narsheedha Beegum. "Malayalam Noun and Verb Morphological Analyzer: A Simple Approach." International Journal of Software & Hardware Research in Engineering, ISSN 2347-4890 (2014).

[11] Varma, AR Rajaraja, and Putusseri Ramacandran. Keralapaniniyam