A Novel Approach For Morphing Telugu Noun Forms Using Finite State Transducers

Sneha. D. L
MTech, SCSE
VIT University

Dr. G. Bharadwaja Kumar
Assistant Professor, SCSE
VIT University

Abstract

Though a number of attempts have been made from the last 30 years, it has been a cumbersome task to design good morphological analyzer for Dravidian languages. Over 200 million people in Asia speak Dravidian languages and the main Indian languages like Tulu, Kannada, and Telugu fall under this category. This paper describes a study which emphasizes the importance of processing the morphology of the Telugu language and difficulties in designing a good system to perform the morphological tasks. We propose a novel approach on how to perform the computational morphology for Telugu noun forms using finite state transducer techniques.

Keywords:
Natural Language Processing, Morphology, Language Analysis, Telugu Noun Forms, Finite State Transducer, Finite State Automata, Foma.

1. Introduction

Morphology is a branch of Linguistics that describes and analyzes the patterns of word formation, word structure and does identifying, analyzing and describing the structure of both the inflectional and derivational morphemes in a given language. Morphological analysis of a wide range of languages has started to attract the attention of more number of linguistics because it's expected to be one of the good means to identify the strategies of the speaker and it's said that to know the way how human brains work and process a given language, the knowledge of Morphology is necessary. It can be implemented efficiently using finite-state technologies based on finite-state transducers. The efforts to explore weighted finite-state transducers for natural language processing are one of the major ongoing research trends today.

The main goal of this research is to provide a well defined architectural framework of a simple and accurate system for building a morphological analyzer for any language with Finite state transducers using Telugu as an example.

This paper is organized as follows. In Section 1, the importance of morphology for any given language and the aim of this research is briefed. In Section 2, the previous contributions in this field are described. Section 3 discusses relevant facts and difficulties about morphology in Telugu language. In Section 4, the methodology, linguistic knowledge representation and resources required for building the morphological analyzer are explained. Brief explanation of the role of Finite State Transducers in building a Morphological analyzer is also described. We explain the algorithm and design further in Section 5 and propose our approach for morphology of Telugu noun forms. Finally, In Section 6, we draw conclusions by enumerating the simplicity and the accuracy of this system approach and how it solves the questions that were raised above.

2. Related Works

There are number of research methods for developing a Morphological system for language. [1] Presents the machine learning approach for generating Telugu Morphological forms. Irrespective of its inflectional and derivational morphological formation, [2] Analyses and generates Telugu Morphological words. [3] Proposes the novel approach for Performance improvement of the existing Telugu Morphological Analyzer by extracting all the possible decomposition. There are various tools available for constructing morphological analyzers using FSTs; the best ones are developed at Xerox by Karttunen [4] and by Chanod [6]. Another framework for research in finite-state automata which is largely compatible with the Xerox/PARC finite-state toolkit is developed by Hulden [4].

3. Telugu Morphology

Telugu is the second largest spoken language in India and is the official language of Andhra Pradesh. Because

of the diverse strategies present and the high density of words formed, the morphologically rich Telugu language is cumbersome to analyze. From the computational point of view, the morphology has become very important, because it plays a major role as an important component of NLP task. The main challenge in morphing Telugu Noun forms is that, the words are formed by concatenating suffixes with one or more stem/roots.

Type	Description
Countable Nouns	These are quantified directly by numerals. Ex: A-vu , A-vu-lu
Uncountable Nouns	These indicate qualities, ideas, unbounded mass etc that cannot be quantified directly by numerals. Ex: u-ppu
Masculine Nouns	These display grammatical relations associated to male beings. Ex: rA-mu-Du
Feminine Nouns	These display grammatical relations associated to female beings. Ex: sI-ta
Neuter Nouns	These display grammatical relations neither associated to male nor female beings. Ex: Gani-Tham
Proper Nouns	These indicate individual entities, such as names of organizations, places or persons. Ex: Chen-nai

Table - 1. Major Classification of Telugu Noun Forms**.**

Telugu nouns are broadly divided into 3 classes:
(1) Proper or Common nouns
(2) Pronouns
(3) Special types of Nouns viz.., Numerals, Verbal

4. Methodology

Telugu linguistic resources are very inadequately available. We are making use of the orthographic rules to form Telugu noun morphemes which we feed into the finite state transducer using the Lexc scripts. The resources to build the morpho analyzer for any language are described in the Table – 2.

Lexicon	Is a catalogue of the languages words (roots/stems)
Dictionary	Lists the lexicons of a given language in alphabetical order.
Grammar	Is a system of rules which derive meaningful information
Affixes	Viz Suffix, prefix, Infix.

Table – 2.

We are making use of the open source compiler, programming language, and C library- Foma [5] for constructing finite-state automata and transducers. The library contains efficient implementations of all classical automata/transducer algorithms and more advanced construction methods such as the transducers from replacement rules, first-order regular logic, etc are supported. A morphological analyzer can be considered as a black box that translates word forms (such as cars) into a string that represents its morphological makeup, such as run+N+Pl: a plural noun form.

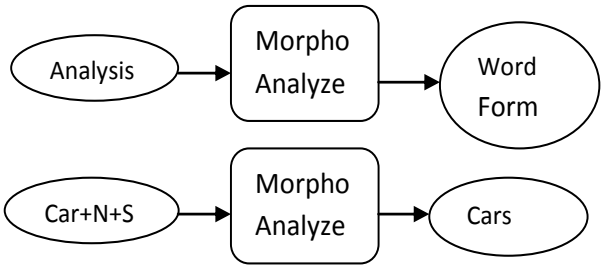


Fig - 1. Morphological Analyzer

The goal is to build a finite-state transducer that accomplishes this string-to-string mapping of analyses to surface forms and vice versa.

4.1 Finite State Transducers

FST can be defined as a finite state machine with an input tape and an output tape. Finite State Transducers play a major role in building a Morpho analyzer for any given language. For a given non recursive application in linguistics, the FSTs are equivalent to the replacement rules. The grammar can be corrected by visualizing the automata using the Graphic tools.

Figure-2 shows an example of a regular expression for generating the Telugu words (PaTTu, TiTTu and KoTTu) using FST. With a simple regex grammar, the FST can be built as shown in Fig - 2.
Syntax: regex [P a — T i— K o] TTu;

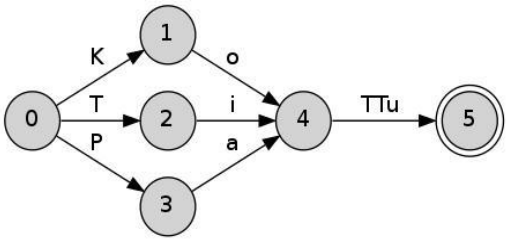


Fig - 2.

In sequential transducers, the use with a given input does not depend on the size of the transducer but only on that of the input. Hence the total computational time is linear in the size of the input and the cost of copying out each output label does not depend on its length.
Fig-3 shows an example of sequential transducers. From the computational point of view, time efficiency is usually achieved by using deterministic automata and the use of finite-state machines considerations reduces the time and space efficiency.

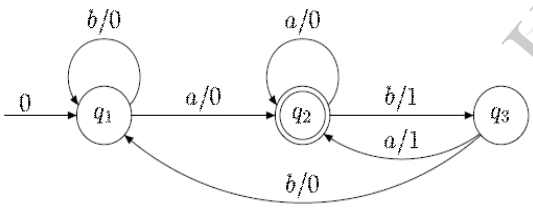


Fig - 3. Sequential Transducer

4.2. Linguistic Knowledge Representation

Morphological processing is represented in terms of lexeme and features. A set of rules governing morphological behaviour are developed, here we do not deal with issues relating to word sense or any semantic predictability. In all Telugu categories, the order
Of clitic affixation is at the right most ends and except adjectives almost all are inflected for clitics. Consider as an example, if we want to generate an inflected form of a root word 'tiNnaDu' - 'has he eaten. The order of suffix is the root word tinu+Du. Insertion of N and Deletion of vowel 'u' are governed by the orthographic rules and the morphological makeup, such as tinu(eat)+V+3p+Sg: describes a verb in the third person singular present tense and the '+' sign indicates the ordering of affixation morphemes.

5. Algorithmic Design and Implementation

We are analyzing the Telugu morphemes based on the orthographic rules. To build the Morphological analyzer, a concatenation rule set has to be generated based on the language rules for combining the affixes. Their main components involved in construction of Finite state transducers are; defining the Lexicon, composing and compiling the orthographic rules set.

5.1 The Lexicon

The lexicon component is a transducer that has the following properties:
(1) Lexicon accepts as input only the valid stems of the language, followed by only a legal sequence of tags.
(2) Produces an intermediate form where the tags are replaced by the morphemes as the output. It may additionally produce special symbols that serve to mark the presence of morpheme boundaries in the output.
As a simple example, we will now look at constructing the analyzer for the plural noun forms in Telugu language. This example is a highly simplified version of the actual morphology. The default rule is of course to just add a 'Lu' as in root+Lu.
The FST lexicon component will perform the following mappings:

Input side:
r a j u +N +Pl
b a S s u +N +Pl
Output side:
r a j u ^ Lu
b a S s u ^ Lu

Here, each complete tag is one separate symbol, a multicharacter symbol (+N, +Pl, etc.) and the spaces in the above show the symbol boundaries. We are using the symbol to mark a morpheme boundary and each letter in the stem is represented by its own symbol (r, a, j, u, etc.). To illustrate this, Fig-4 shows how the lexicon transducer that only encoded these two words would look like.

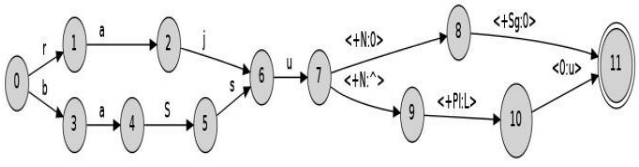


Fig - 4.

5.2 Orthographic Rules

Then there are some stems where care should be taken to form the plural, like for e.g. the plural form of guDDu is guDLu. This can be explained by few morpho-phonological rules that are defined in the Telugu grammar. Table-3 briefs four rules for plural noun formation.

Rule	Singular Form examples	Plural Form Examples
If Stem final [i/e] preceded by [T/NT/ND] then its lost before the plural suffix [-Lu]	kooTi korTu baNDi paNDu	kooTLu korTLu baNDLu paNDLu
If Stem Final has [Di/Du/lu/ru/li/ri] then replace it with 'L' before suffix [-Lu]	kaalu uuru pandiri waakili peraDu	kaaLlu uurLu pandirLu waakiLlu peraDLu
If Stem Final has [Tti/TTu/DDi/DDu] the replace the doubles with single [T/D] before suffix [-Lu]	puTTi guDDu	puTLu guDLu

Table - 3.

So, we want a transducer that translates the given nouns into plural forms using the rules from the above table. At the very least we will need to remove the auxiliary symbol which is used to separate morpheme boundaries. The last rule is defined to remove this. But, there are a few nouns that do not comply with any orthographic rules; such exceptional cases must also be defined. The rules mentioned above can be defined in lexc as shown below:

# RuleOne:	define RuleOne T i 0 -> T || T u 0 -> T || N T i 0 -> N T || N T u 0 -> N T || N D i 0 -> N D || N D u 0 -> N D ;
# RuleTwo:	define RuleTwo D i 0 -> L || D u 0 -> L || l u 0 -> L || r u 0 -> L || l i 0 -> L || r i 0 -> L;
# RuleThree:	define RuleThree T T i 0 -> T || T T u 0 -> T || D D i 0 -> D || D D u 0 -> D;
#CleanUp:	Define Cleanup " ^ " -> 0;

Table – 4.

5.3 Compiling the Grammar

Before compiling the grammar, in the same script that compiles the replacement rules, read in the lexc file, and label it as Lexicon in order to use the label in subsequent regular expressions.

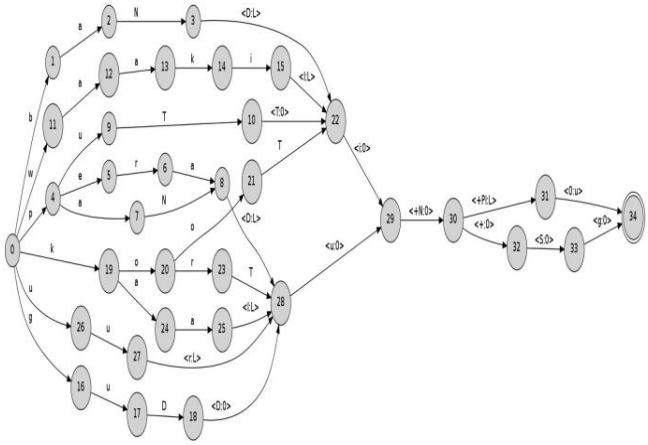
```
read lexc telugu.lexc;
define Lexicon;
```

Having defined and labeled all the necessary rules and the lexicon, these rules can be composed by using regex expressions:

```
regex Lexicon .o.
RuleOne .o.
RuleTwo .o.
RuleThree .o.
Cleanup;

# .o. refers to composition operation
```

Thus, by using the above rules, we can generate the FST as shown in Fig - 5. After developing the analyzer based on the above design steps, it can be saved in a bin file. Finally its performance has to be tested using a huge Telugu Corpus that takes randomly both the



inflected or derived forms. Thus the algorithmic steps for developing the Telugu Morphological analyzer are as shown below in Fig - 6:

Fig – 5.

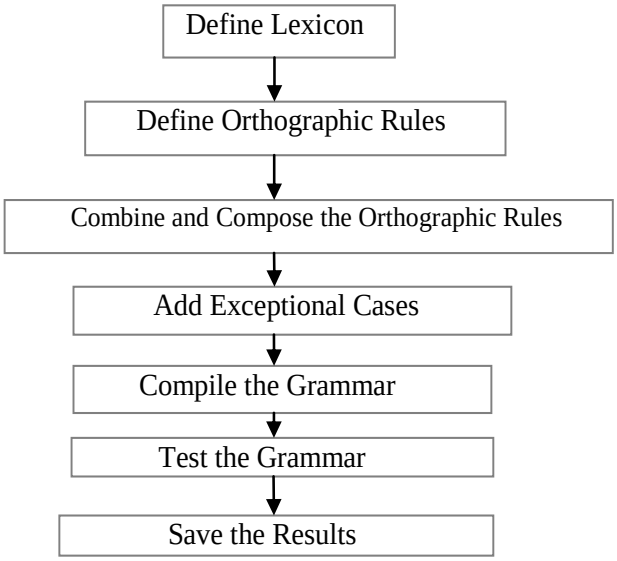


Fig – 6.

6. Conclusion and Future Research

In this paper, we proposed a novel approach to construct a simple and efficient Morphological Analyzer for Telugu Noun forms by incorporating rules in Finite State Transducers. Our approach is Novel because of the following main reasons:

(1) Simplicity of the algorithmic design which helps in building the system faster.

(2) Carefully compiled orthographic Telugu rules using lexc regex scripts that promises accuracy.

(3) Fully supports various formats for specifying the regular expressions and Uni-code.

(4) Since it's largely compatible with the Xerox/PARC finite-state toolkit, the FSA can be viewed graphically.

(5) Fully supports various formats for specifying the regular expressions and Uni-code.

(6) Separate utility (flookup) for applying automata with various strategies and separate C API for constructing and handling automata.

Presently the system is designed to analyze the plural morphemes for Telugu noun forms. Further lines of research and development include:

(1) Completing the development of the proposed idea for defined Telugu noun data set.

(2) Evaluating the performance of the proposed method using huge Telugu Corpus.

(3) Providing an animated Finite State Automata for parsing the given input.

Thus, this paper described a simple framework for designing and building a Morpho Analyzer for Telugu noun forms using the Telugu orthographic rules set with FST.

7. References

[1] Madhavi Ganapathiraju and Lori Levin. TelMore: Morphological Generator for Telugu Nouns and Verbs, Mellon University, Pirrsburgh, PA 15213, USA.

[2] Uma Maheshwar Rao G, Amba Kulkarni P and Christopher Mala. A TELUGU MORPHOLOGICAL ANALYZER, .Center for Applied Linguistics and Translation Studies, University of Hyderabad, India.

[3] K.V. N. Sunitha and N. Kalyani. A Novelapproach to Improve rule based Telugu Morphological Analyzer, World Congress on Nature and Biologically Inspired Computing, 2009.

[4] Karttunen L. Finite-State Lexicon Compiler. Xerox Palo Alto Research Center (1993) Palo Alto, California.

[5] Mans Hulden. Foma: a finite-state compiler and library. University of Arizona.

[6] K. Beesley. Arabic finite-state morphological analysis and generation, volume 1. Proceedings of the 16th International Conference on Computational Linguistics (COLING-96), pp.8994, Copenhagen, Denmark, 1996.

[7] Krister Lindn, Miikka Silfverberg, and Tommi Pirinene. HFST Tools for Morphology An Efficient Open-Source Package for Construction of Morphological Analyzers. Department of General Linguistics, University of Helsinki, Finland.

[8] Mohri, M., Riley. M. An efficient algorithm for the n-beststrings problem. Pro-ceedings of the International Conference on Spoken Language Processing 2002.

[9] Beesley. K and Karttunen. L Finite State Morphology. CSLI Publications (2003).

[10] Mohri. M. Finite-state transducers in language and speech processing. Computational Linguistics 23(2), (1997).