Bidirectional Analyzer and Generator Tool for Kannada Nouns

Bhuvaneshwari C Melinamath 1
1 Dept. of Computer Science
B.L.D.E.A.'s Engg College and Tech.
Bijapur 586103. Karnataka, India
1 bmelinamath@yahoo.co.in,

Abstract

This paper explores a computational model for Kannada nouns. This is a single tool performing analysis and generation in reverse direction. Kannada has rich morphology with inflections -vibhakti, derivations and compound formation. In depth study of words and its structures is done using this tool. Through study of noun from computational linguistics point of view is important. An exhaustive version of the morphological analyzer and generator tool for nouns using FST is described here. Morphological analyzer/generator is designed to interface with a knowledge-based machine translation (MT) system. Our system uses a hierarchy (tree) structure to relate various morphological features. Morphological analyzer/generator for morphologically complex and agglutinative language like Kannada is highly challenging. The major types of morphological process like inflection, derivation, and compounding are handled in this system. All these developments have worked on the basis of syntactic structure of Kannada language. A robust analyzer is useful for language learning as well as for machine translation applications. The tool is tested against 10000 words and has 98% recognition success rate.

1. Introduction

Morphological analyzers (MA) are essential parts of many natural language processing systems such as machine translation systems (MT) and spell checkers [2], word net [3] Information retrieval systems, Tagged corpus Generation (TCG). The objective of the present work is to build a computational model for analysis and generation of Kannada language nominal words. Morphological analysis reads the inflected surface form of each word in a text and writes its lexical form consisting of canonical form of the word and a set of tags showing its syntactic category and morphological characteristics. The analyzers rely on two sources of information: a dictionary of valid lemmas of the language and a set of rules for inflection handling. Finite states transducers (FST) are a most efficient approach to morphological analysis [8, 1, and 3] a class of finite state automata. There are a number of tools for the construction of FST based morphological analyzers the best known being developed at Xerox for a review in Spanish on finite state morphology. In this work a FST based morphological analyzer is developed. The number suffix attaches to the noun root, followed by case suffix. Postpositions follow the case and clitics. Kannada morphology is largely concatenative; our tool handles prefixation and suffixation but not infixation, as Kannada does not support infixation.

Our first attempt was to use the system to analyze Kannada noun later extend to generation morphology. There is no significant duplication of rules except while handling genitive suffix marker 'a' for noun while same suffix is used as negative marker in verbs and also as imprecate marker used for cursing in verbs. From a practical perspective, duplication also presents a maintenance problem. When we began by extending the morphology system to handle Kannada noun compounds, echo-words and the broken plurals of nouns [2, 6, and 5] in this paper, we begin by sketching the original morph system, but we focus on the current extensions required to accommodate Kannada morphology and on the integration of the morphology system with MT system. We conclude by briefly comparing our approach to other treatments of Kannada morphology and describing future work.

2. Computational Model

The morphological generators/analyzers are based on finite state transducers; in particular, we use string or pattern transducers unlike letter transducers by Roche & Schabes [6]. Any finite-state transducer may always be turned into an equivalent letter transducer. Instead of transition on letters we have transitions on sequence of letters i.e. strings generally valid suffixes in the language. The transducer is defined as $T = (Q, L, \delta, qI, F,)$ where Q is a finite set of states, L is a set of transition labels, $qI \in Q$ the initial state, $F \subseteq Q$ the set of final states, and $\delta: Q \times L \rightarrow 2Q$ the transition function. The set of transition labels is $L = ((\Sigma \cup \{\epsilon\}) \times (\Gamma \cup \{\epsilon\})$where Σ is the alphabet of input symbols, Γ the alphabet of output symbols, and ϵ represents the empty symbol. According to this definition, state transition labels may therefore be of four kinds: $(\sigma: \gamma)$, meaning that symbol $\sigma \in \Sigma$ is read and symbol $\gamma \in \Gamma$ is written; $(\sigma : \epsilon)$, meaning that a symbol is read but nothing is written; $(\epsilon : \gamma)$, meaning that nothing is read but a symbol is written; and $(\epsilon : \epsilon)$ means that a state transition occurs without reading or writing. The last kind of transitions are not necessary neither convenient in final FSTs, but may be useful during construction. It is customary to represent the empty symbol ϵ with a zero ("0"). A letter transducer is said to be deterministic when $\delta: Q \times L \rightarrow Q$. Note that a letter transducer which is deterministic with respect to the alphabet $L = ((\Sigma \cup \{\epsilon\}) \times (\Gamma \cup \{\epsilon\})$may still be non-deterministic with respect to the input Σ. A string $w' \in \Gamma$□ is considered to be a transduction of an input string $w \in \Gamma$□ if there is at least one path from the initial state qI to a final state in F whose transition labels form the pair $w : w'$ when concatenated. There may in principle be more than one of such paths for a given transduction; this should be avoided, and is partially eliminated by determinization. On the other hand, there may be more than one valid transduction for a string w (in analysis, this would correspond to lexical ambiguity; in generation, this should be avoided). In analysis, the symbols in Σ are those found in texts, and the symbols in Γ are those necessary to form the lemmas and special symbols representing morphological information, such as <noun>, <fem>, <first personp1. Second person p2. Third person p3>, etc. In generation, Σ and Γ are exchanged. The general definition of letter transducers is completely parallel to that of non-deterministic finite automata (NFA) and that of deterministic letter transducers, parallel to that of DFA; accordingly, letter transducers may be determinized and minimized (with respect to the alphabet L) using the existing algorithms for NFA and DFA as in Hopcroft & Ullman [9]. Transitions labeled $(\epsilon:\epsilon)$ may be eliminated during determinization using a technique parallel to ϵ -closure. I.e. NFA with epsilon moves, which allows transformation to a new state without consuming any input symbols.

2.1. Theories in Kannada Morphology

The word formation process in Kannada encompasses 3 major types of morphology. Pronoun morphology, Inflectional morphology and derivational morphology [4]. Pronoun morphology is study of grammatical classification of pronouns. For example nammindaleena (is it from us only?) indicates personal, 1st person and plural number with ablative case emphatic clitics followed by interrogative clitics at the second level. Inflection morphology (IM) is combination of root word with grammatical morphemes, usually resulting in a word of same class as the original stem with some syntactic information like number, person etc., Derivational morphology (DM) is combination of a stem with grammatical morphemes, resulting in a word of different class. Our analyzer relates bidirectionally a lexeme and a set of linguistic features to a surface word form through a set of transformations. In Kannada all words end in vowel, the behavior of the word is different based on its endings. All categories except adjectives are inflected for clitics; order of clitic affixation is at the right most end. Consider an example below Insertion of 'y/n/d/v/l', and deletion of vowel ‘u’ is governed by orthographic rules [7]. In Kannada, unlike some languages, two vowels do not coalesce when they occur in adjacent morphemes; rather a glide is inserted between them. For example, the FS for analysis the Kannada word ‘maradiMdaleenaa’ (is it from the tree?) would be: maradiMdaleenaa=mara |+ iMda+ee |+aa. The '+” sign indicates the ordering of affixation morphemes. ((root "mara") (cat noun,) (common) (case ablative) (‘ee’ emphatic marker) ('aa' inclusive clitic) (‘d’,'l', and ‘n’ glides insertions) (number sg) (gender neuter)).

2.2. Aspects of Nominal Inflection and Irregular Forms.

As Kannada has rich morphology there are about 256 word forms, inflection morphemes for one Kannada noun the detailed classification of nominal suffix is shown in table 1.

All morphology is not regular. Irregular words are those which don't obey the normal rules of inflection. FS is matched against the features defining each sub tree in the MFH until a leaf node is reached. The analyzer then checks the irregular form in the lexicon for an entry indexed by the value of the root feature and the name of the node and returns it, if there is one. Otherwise it attempts to apply the transformational rule attached to the leaf node. Some irregularities like plural form of children 'makkaLu' is listed in lexicon, and the stem denoting kinship like amma, akka etc., follow different rule of addition of 'aMdiru' to form plural formation instead of plural formation rule of adding suffix 'ru' to form animate plurals. Similarly terms indicating honorific agrees with plural third person indicator with its subject verb agreement. Some other words like janaru,”people”, indicate plural forms, which do not have corresponding singular form in the language. Such irregularities are listed in dictionary. The process of combining morphemes involves a number of orthographic rules that modify the form of created word, so it is not a simple interleaving or concatenation of its morphemic components.

2.3. Derivation of Echo Words from Nouns.

Consider a sentence “peTege hoogalu kaarugiiru” iddare cennagiruttade. (“It would be better if there is a car to go to market”). Kaarugiiru is an echo word. kaar has meaning of car but giiru is not meaningful word. This is specialty of echo words. Echo words are formed by the substitution morpheme 'gi'/'gii' with first akshara of the word if it long vowel then gii is inserted otherwise gi. Words which begin with p must have only gi/gii and vice versa. Although 'gi'/'gii' is the normal morpheme, 'pa'/'paa' occurring as an allomorph is still used in many instances in place of gi/gii. For example (a) uuTa-giiTa, or uuTa-paaTa , “meals”. (b) paaTha-giiTha, “Lessons”. Here first words are regular lexical item of vocabulary second words like giiTa, gistaka are an echo words indicating some sense, such words are actually does not carry any lexical meaning and also not part of lexicon. Echo words occur together without any space or hyphen between them.

Sample algorithm for echo word generation

```
Derivation of_Echoword (stem, cat)
/*cat should be noun, verbs, adverbs, proper names*/
 If first akshara of the word is 'gi'
     if it is short vowel then
     replace first akshara with 'pa' to form first akshara of second word.
     Else
       replace first akshara with 'paa' to form first akshara of second word
Else /*other than gi/gii*/
   if first akshara is short vowel
 replace first akshara with 'gi' to form first akshara of second word.
   Else
 replace first akshara with 'gii' to form first akshara of second word
```

Figure 1.Rules for generating echo words

Table 1. Kannada Case System

Vibhakti/Case	Suffix	Case Relations
Nominative	uu	subjective
Accusative	annu	objective
Ablative	iMda	Instrumental
Dative	ige,kke,ge	objective
Genitive	a	Non case
Locative	alli	locative

3. Derivational Morphology for Noun

We can derive adverbs from noun, by adding suffix aMte" (like), and adjective by adding aMtaha (similar), and possessive form of noun which acts as adjectives by adding suffix "a". Our tool uses a specific FVP or set of FVPs to represent special cases requiring special rules. These FVPs distinguish leaf-nodes from a parent node. In Kannada we have productive rule of deriving human nouns by adding third person pronoun suffixes to descriptive adjective like dodda+avanu= doddavanu, dodda "big" is a adjective, avanu "he", indicating younger brother, this kind of derivation is handled through FSM, and the adjective is dodda (big) is the lexicon entry [4].

4. Results and Experiments

We conducted the experiments to evaluate the system for nouns. The system was evaluated based on the parameters of precision and recall. Precision. For the purpose of this experiment, precision is defined as the number of true positives (number of words correctly generated) divided by the total number of true outputs, i.e., the total number of inputs that should be correctly generated.

Recall. For the purpose of this experiment, recall is defined as the number of true positives (number of words correctly generated) divided by total number of positives.

F-measure. F-measure is defined as F = ((2 * Precision * Recall)/ Precision + Recall)

We evaluated the following source using lexical knowledge dictionary of 30000 words compiled by us using hierarchical tag set. And DoE CILL Text corpora are used.

Table 2. Input words with noun roots spanning different paradigms and attributes

Nouns (1000 input words) Results		
True positives	9600	
use Negatives	180	
False Positive	200	
False Negatives	20	
Precision	98%	
Recall	97%	
F-measure	97%	

The generator/analyzer can work within the framework of the MT system [6]. Echo word can be inflected for case and clitics just like noun. Some criteria must be adopted for handling echo words, when generator are used as part of MT systems, since languages like English do not support echo words.

This tool is an extension of the generator. We have tried to perform analyzer and generator as single tool, but we observe generator becomes too general when applied for some inflection on verbs which is ongoing work. Taking measures to reduce generalization is our next issue.

References

[1] Martin Kay, “Nonconcatenative finite-state morphology,” In Proceedings of the Third Conference the European Chapter of the Association for Computational Linguistics, pages 2–10, 1987.

[2] Beard, R.: Lexeme-Morpheme Base Morphology: A General Theory of Inflection and Word Formation. State University of New York Press (1995)

[3] Kiraz, G.: Multi-tape Two-level Morphology: A Case study in Semitic Non-Linear Morphology. In: Proceedings of COLING’94, Vol. 1 (1994) 180-186.

[4] Bhuvaneshwari.C. Melinamath,” A Robust Morphological Analyzer to Capture Kannada noun Morphology” Proc. International conference on Future Information Technology. IPCSIT vol.13 (2011), Singapore.

[5] Nizar Habash, 2004, “Large scale lexeme based Arabic morphological generation,” In Proceedings of Treatment Automatique du Language Natural (TALN-04). Fez, Morocco.

[6] Roche, E and Y Schabes, “Introduction. In finite state language processing,” Ed . By E. Roche and Y Schabes, 1-65. Cambridge, Mass. MIT Press, 1997.

[7] Bhuvaneshwari.C. Melinamath,” A Morphological Generator for Kannada based on Finite State Transducers” Proc. ICECT 2011, V1-312 – 316, Kanyakumari, India.

[8] M. Mohri.: Finite-state transducers in language and speech processing. Computational Linguistics, 23(2):269–311 (1977).

[9] Hop croft J.E and J D Ullaman, Introduction to automata theory languages and computation. Reading MA: Addition Wesly, 1979.