# Development of Prototype Morphological Analyzer for the South Indian Language of Kannada

T.N. Vikram and Shalini R. Urs

International School of Information Management, University of Mysore, Manasagangotri,
Mysore-570006, Karnataka, India
{shalini,vikram}@isim.ac.in

**Abstract.** A prototype morphological analyzer for the south Indian language of Kannada is presented in this work. The analyzer is based on Finite state machines and can handle 500 distinct Noun and Verb stems of Kannada. The morphological analyzer can simultaneously serve as a stemmer, part of speech tagger and spell checker and hence it becomes a very efficient tool for content management.

**Keywords:** Kannada Morphology, Finite State Machine, Kannada Content Management, Natural Language Processing.

## 1 Introduction

The onset of localization of the content has capacitated the penetration of internet into those regions which do not speak English, particularly Asia. People can read and post things in their own native languages now. However, the current capabilities of the localized edition of internet is very limited. Key word based searching for the local languages is yet to be developed. Text categorization, summarization and retrieval has not been achieved in most of the Asian languages due to the lack of the essential stemming algorithms which are language specific. Similarly automatic translation of the pages to English or any other language is facilitated only if there is an efficient Part of Speech tagger(POS) [22]. As in the case of a stemming algorithm, most of the Asian languages also lack POS taggers for their respective languages. This can be addressed by developing a morph analyzer for that given language. A morph analyzer outputs the stem, the POS tag and affix for any given word. As a result the morph analyzer can be used for both stemming and part of speech tagging simultaneously.

In view of this, we have attempted to develop a prototype Kannada Morph Analyzer. Kannada is the official language of the south Indian state of Karnataka, with about 44 million speakers. Though a language of rich literary history, it is resource poor when viewed through the prism of computational linguistics. There are hardly any attempts apart from the work of Sahoo and Vidyasagar [5] where a Kannada WordNet is attempted and a Kannada Indexing software prototype by Settar [6]. Both of them are highly constrained by the lack of a morphology analyzer. Unlike English where most the morphotactic changes do not bring about change in spellings, Kannada words change spellings when the stems are inflected, which adds to the complexity of developing the morph analyzer. The analyzer is based on Finite state machines and can handle 500 distinct Noun and Verb

D.H.-L. Goh et al. (Eds.): ICADL 2007, LNCS 4822, pp. 109–116, 2007.

stems of Kannada. The morphological analyzer can simultaneously serve as a stemmer, part of speech tagger and spell checker simultaneously, and hence it becomes a very efficient tool for content management.

The paper is organized as follows. In Section 2 we briefly describe the state of the art in morphology analysis of various languages. Language specific morphology for Kannada is explained is Section 3. In Section 4 we explain the development of the proposed morph analyzer for Kannada. Finally we conclude this work with some discussion in Section 5.

## 2 The Current State of the Art in Morphological Analysis

The morphological analysis for English is far more advanced than any other contemporary languages [1]. Some recent advances in stemming for Germainc and other European languages can be found in Braschler and Ripplinger [7]. A comparative analysis of the various stemming algorithms for nine European Languages is presented in the survey report by Tomlinson [8]. A few stemmers for Asian languages are also proposed in the literature. Lee [9] has proposed a lexical analyzer and stemmer for Korean. It is implemented using finite state machine. A Chinese-English cross language information retrieval based on Chinese stemming is proposed in Min et al [10]. But stemming does not play a very crucial role in Chinese and Japanese because noun phrases never undergo morphotactic changes [2]. A few attempts for Malay language morph analysis is also seen [3].

Relatively little literature is available for Indic languages. An unsupervised morphology learner for Assamese language is proposed by Sharma et al. [11]. An automatic spell check for Assamese is proposed by Das et al. [12]. A Morph analyzer for Oriya has been proposed in Mohanty et al. [13], which works on the paradigm of finite state machines. A few prototype morph analyzers for Tamil, Bangla, Oriya, Assamese and Manipuri has been attempted [14].

The future of any content management activities in Kannada relies on the language technologies like spell checker development, POS tagger and stemmer. A morph analyzer serves as a spell checker, POS tagger and stem identifier simultaneously and hence this works assumes importance. To the best of our knowledge there is no research literature available with regard to the development of a morphology analyzer in Kannada, and hence this work assumes importance. Kannada has 38 basic character. Also 330 conjuncts are formed due to combination of vowels and consonants. Kannada has 100,000 basic stems and more than a million morphed variants formed due to more than 5000 distinct character variants. We report here the development of a finite state Kannada morphology for Nouns and Verbs. It has been implemented on the AT & T FSM Toolkit [17]. The nuances of verb and noun morphology of Kannada are explained in the section to follow.

## 3 Kannada Morphology

In Kannada, the derivation of words is either by combining two distinct words or by affixes. During the combination of two words spelling changes might occur. Eg: mugilu (Sky) + eVttara (High)= mugileVttara (Sky High). Word combination occurs in two ways in Kannada, namely *Sandhi* and *Samasa*[15]. However we do not handle compound formation morphology in this work. The other method in which word formation happens is through affixes.

Kannada inherits many of the affixes from Sanskrit. Most of the Kannada affixes are inflectional suffixes called *vibhakati*s[15]. Spelling changes occur in a large number of cases with the application of suffixes. Kannada inherits 20 prefixes (*upasarga*s) from Sanskrit. Prefixes in many cases change the meaning of the words in a way that the derived words may be a treated as root words themselves.

### 3.1 Nouns

Nouns represents the gender, rationality, case and number in Kannada. The nouns ending with –anu and –aLu are identified as masculine and feminine respectively. Kannada nouns in their singular for do not have any markers attached to them. –gaLu, –a, –aru are generally considered as the plural markers. Eg: The root *bAlaka* (Boy) is pluralized as *bAlakaru* (Boys), *maneV* (House) is pluralized as *maneVgaLu* (Houses).

Similar case marking exists in Kannada as in other Dravidian languages. The case markers for the corresponding cases is given in Table 1. However they cannot be merely concatenated to the roots.

**Table 1.** Case markers for nouns [18]

| Case | Marker |
|---|---|
| Nominative | -0(u,nu,lu,ru) |
| Accusative | -annu, -vanna, |
| Genetive | -a |
| Dative | -ge, -ige, -akke, -kke |
| Locative | -alli, -yalli, |
| Instrumental/Ablative | -inda, YiMda |
| Vocative | -ee / vowel length |

**Table 2.** Inflected noun and its meaning when different markers are concatenated

| Inflected Nouns | Meaning in English | Type of Inflection made on the Noun by markers |
|---|---|---|
| hani | Drop | - |
| hani-gaLu | Drops | Plural |
| hani-yiMda | From the Drop | Singular + Ablative |
| hani-geV | To the Drop | Singular + Dative |
| hani-ya | Of the Drop | Singular + Genitive |
| hani-yalli | In the Drop | Singular + Locative |
| hani-yannu | The Drop | Singular + Accusative |
| hani-galYiMda | Because of Drops | Plural + Ablative |
| hani-galYigeV | To the Drops | Plural + Dative |
| hani-galYa | Of the Drops | Plural + Genitive |
| hani-galYalli | In the Drops | Plural + Locative |
| hani-galYannu | The Drops | Plural + Accusative |

Nouns that terminate with vowels like (eV, i, u, A) are appended with an *a*, preceded by morphophonomically inserted y or eV. These case markers are strictly for singular cases. For plural cases the markers themselves undergo certain morphotactic changes. An example is considered in Table 2, which illustrates the inflection of the noun *hani*(Drop), with singular and plural case markers.

For convenience the stems are separated from the suffix with a hyphen in Table 2. It shall be observed in Table 2, that the morphophonemic *y* is inserted in the Singular Locative and Accusative cases.

### 3.2 Verbs

Kannada verbs occur in both Finite and non-Finite form. Most of the verb stems are in non-finite form for which tense, markers and grammatical forms are added. The singular polite form of a verb is generally obtained by adding +i, and the plural polite form is obtained by adding +ri. For example consider the word *noVdu* (See), the singular and plural polite forms thus obtained are *noVdi* (Please See) and *noVdiri* (Please be so kind enough to see) [24].

**Table 3.** Inflected verb and its meaning when different markers are inflected

| Inflected Verbs | Meaning in English | Tense | PNG |
|---|---|---|---|
| nadeV- yuttA-ne | He will walk | Future Progressive | 3SM |
| nadeV - yuttA-lYeV | She will walk | Future Progressive | 3SF |
| nadeV - yuttA-re | They will walk | Future Progressive | 3P- |
| nadeV - yuttidda-ne | He is walking | Present Progressive | 3SM |
| nadeV - yuttidda-lYeV | She is walking | Present Progressive | 3SF |
| nadeV - yuttidda-re | They are walking | Present Progressive | 3P- |
| nadeV - yuttidda-nu | He was walking | Past Progressive | 3SM |
| nadeV - yuttidda-lYu | She was walking | Past Progressive | 3SF |
| nadeV - yuttidda-ru | They were walking | Past Progressive | 3P- |
| nadeV - yuttiddI-ya | You are walking | Present Progressive | 2S- |
| nadeV - yuttidde-ne | I am walking | Present Progressive | 1S- |

The important aspect of verb morphology is the Person-Number-Gender (PNG) and the tense marker concatenated to the verb stems. Unlike English, Indic languages add Gender information to verbs also. The general syntax for this is Verb stem + Tense Marker + PNG Marker. The morphotactic changes that occur in a verb when tense information is added is highly subjective. For example the morphotactic changes in the verb *nadeV* (To Walk) is given in Table 3. Note that the PNG marker in Table 3 is as follows, (1/2/3)(S/P)(M/F/-), where 1/2/3 denote $1^{st}$ , $2^{nd}$ , $3^{rd}$ Person, S/P indicate Singular/Plural and M/F/- indicates Masculine, Feminine or Neutral Gender.

The development of the proposed prototype morph analyzer for Kannada is illustrated in the next section.

## 4 Construction of Morph Analyzer for Kannada

We employ finite state machines (FSM) for the development of Kannada morph analyzer. The primary attraction for using a FSM, for the purpose of developing a morph analyzer is its speed and efficiency. Many natural language processing techniques routinely employ FSM for shallow parsing, syllabification, tokenization and spell checking[16]. When compared to many unsupervised methods of learning morphology, an FSM based morphology development is a more tedious process because all the rules and the morphophonemic changes have to be hard coded. But the major advantage is that once a verb or noun paradigm is identified it is just a matter of identifying the stems with which it can be concatenated. The formal grammar for Kannada nouns and verbs thus identified and are given in Fig. 1 and Fig. 2 respectively.

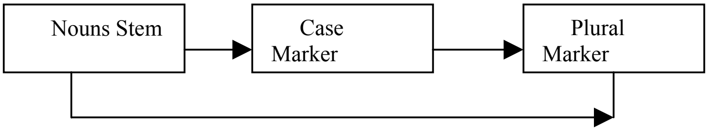

**Fig. 1.** A Formal Grammar for Kannada Nouns

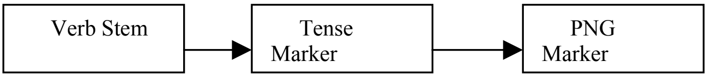

**Fig. 2.** A Formal Grammar for Kannada Verbs

Morphological analysis with FSM is based on the assumption that the mapping of the words to their underlying analysis forms a regular set, and there is a regular relation between these sets. In languages where morphotactics is morph concatenation only, FSM's are straight forward to apply. Handling non-concatenative or partially concatenative languages is highly challenging [19].

The development of the Morph analyzer for Kannada is hindered by the lack of publicly available dataset. Hence we have created a dataset of 500 distinct noun and verb stems. The dataset is in the Roman transliterated form and we have used the ITRANS [23] prescribed Kannada to Roman character mapping. One of the major difficulties in developing any language analysis tools for Indic languages are that the number of diacritics and compound character symbol totals to about 80,000 in number [21]. Unlike in English it is just 52 distinct symbols, the upper and lower cases of the 26 alphabets.

The dataset that we have created has 1014 distinct Kannada character symbols. With this we have implemented a Finite State Transducer(FST) on the AT&T [17] Toolkit. Transducer is a kind of machine, which translates a given input into a specified output [20]. For a given input the designed transducer outputs the stem, part of speech and case marker. A transducer is created for stem with all its morphotactic changes. The transducer for the noun *hani*, illustrated in Table 2 is given in Fig. 3.

The circles in the Fig. 3 indicate the states and the arrows indicate the transitions. Each transition is assigned a symbol of form *x:y*, where *x* is the input symbol and *y* being the corresponding output symbol. In order to maintain time efficiency, the transducer is

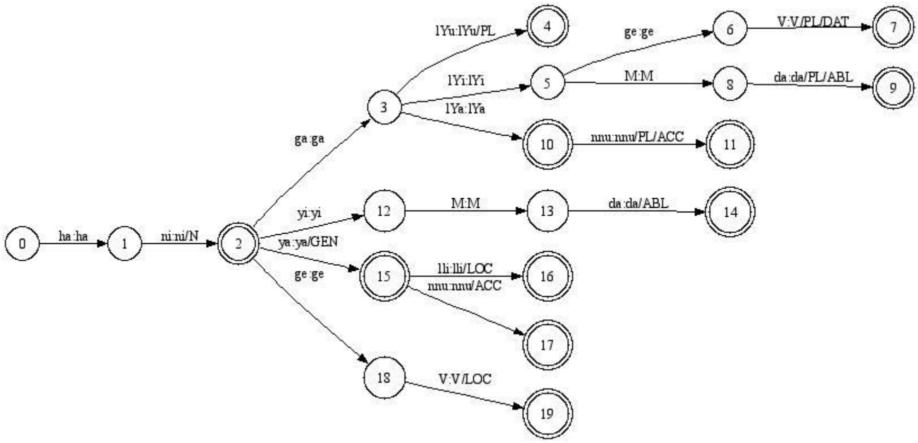

**Fig. 3.** FST for the noun ‘‘hani’’. Legend-> N: Noun; PL: Plural; DAT: Dative; ABL: Ablative; ACC: Accusative; LOC: Locative; GEN: Genative.

designed as a deterministic finite automata (DFA). DFA has minimal number of redundant transitions and hence the complexity of the network is reduced [20].

Consider the word ‘*haniyiMda*’ of the stem *hani* illustrated in Table 2, as the input given to the transducer given in Fig. 3. The transducer produces the output as ‘*hani*/N *yiMda*/ABL’. N stands for noun and ABL stands for ablative. The stem *hani* is thus concatenated with the POS tag: N and *yiMda* is concatenated with the case marker: ABL. The stem *hani*/N is output from the transitions 0, 1 and 2, and the case marker *yiMda*/ABL is output from the transitions 2, 12, 13 and 14.

Likewise an FST is written for each of the individual Noun and Verb stems to accept them. An FST accepts a query only if the word is contained within its transitions and not otherwise. During query the word has to be subjected to parsing by all the developed FSTs, and the query will be accepted by only one of the FST which contains the word in its transitions. However passing a query input string to obtain its part of speech and stem for all the developed FSTs is unwieldy. This is overcome by merging all the developed FSTs and making it a single unified FST. AT & T toolkit provides the necessary commands to merge individual FSTs.

The developed prototype analyzer has the capability to handle around 7000 distinct words from 500 distinct noun and verb stems. But it is far from a being full fledged morph analyzer as pronoun and adjective morphology have not been included in this work.

## 5 Contributions and Conclusion

We have developed a prototype morph analyzer for Kannada for the very first time in the literature. The developed morph analyzer can be used as a spell checker, POS tagger and stemmer simultaneously. This serves as an efficient tool for the preprocessing activities of

Kannada document digitization and content management. The performance of the existing OCRs for Kannada can be improved by modifying the morph analyzer to a spell checker, thereby correcting the mistakes, which the OCR has incurred [4]. As it also serves as a stemmer, Kannada document summarization and classification is made possible, which has not been attempted yet. It also serves as a tool for language translation because it identifies the POS tag. POS tag identification is the first pre-processing for machine translation. Our future goal is to develop a morph analyzer, which can handle words from 15,000 distinct stems from the current capability of 500 stems.

## Acknowledgement

This paper is in continuation of the project carried out by T. N. Vikram at the Microsoft NLP Summer School (May 3-17, 2007) at the Indian Institute of Science (IISc), Bangalore, India. He would like to thank his team mates V.N. Manjunath Aradhya, S. Noushath and team mentor S. Baskaran for the support rendered during the project days at IISc. Support from MSR India is appreciated.

## References

1. van Rijsbergen, C.J., Robertson, S.E., Porter, M F : New models in probabilistic information retrieval, British Library, London (1980)
2. Zhou, Y., Qin, J., Chen, H., Nunamaker, J.F.: Multilingual Web Retrieval: An Experiment on a Multilingual Business Intelligence Portal. Digital Object Identifier (2005), doi:10.1109/HICSS.2005.450
3. Idris, N., Syed, S.M.F.D.: Stemming for Term Conflation in Malay Texts. International Conference on Artificial Intelligence (IC-AI 2001) (2001)
4. Ma, Q.: Natural language processing with neural networks. Language Engineering Conference, pp. 45–56 (2002)
5. Sahoo, K., Vidyasagar, E.V.: Kannada WordNet - A Lexical Database. TENCON Asia Pacific, pp. 1352–1356 (2003)
6. Setter, S., Goswami, S., Abhishek, H K.: Indexing software for Ancient Kannada Books. Language Engineering Conference (2002)
7. Braschler, M., Ripplinger, B.: How Effective is Stemming and Decompounding for German Text Retrieval? Information Retrieval, 291–306 (2004)
8. Tomlinson, S.: Lexical and Algorithmic Stemming Compared for 9 European Languages with Hummingbird SearchServer$^{TM}$ at CLEF 2003. pp. 286–300 (2003)
9. Lee, C.Y.: Local grammar based lexical analyzer for Korean language. In: Proceedings of VEXTEL (1999)
10. Min, J., Sun, L., Zhang, J.: ISCAS in English-Chinese CLIR at NTCIR-5. In: Proceedings of NTCIR (2005)
11. Sharma, U., Kalita, J., Das, R.: Unsupervised learning of morphology for building lexicon for a highly inflectional language. ACL SIGPHON, 1–10 (2002)
12. Das, M., Borgohain, S., Gogoi, J., Nair, S.B.: Design and implementation of spell checker for Assamese (2002)

13. Mohanty, S., Santi, P.K., Adhikary, K.P.D.: Analysis and Design of Oriya Morphological Analyser: Some Tests with OriNet. In: Proceeding of symposium on Indian Morphology, phonology and Language Engineering, IIT Kharagpur (2004)
14. http://tdil.mit.gov.in/TDIL-OCT-2003/morph%20analyzer.pdf]
15. Hiremath, R.C.: The Structure of Kannada. PhD Thesis. Karnatak University (1961)
16. Amsalu, S., Gibbon, D.: Finite state morphology of Amharic. Workshop on RNLAP (2005)
17. http://www.research.att.com/ fsmtools/fsm/
18. Sharada, B.A.: Transformation of Natural language into an indexing language: Kannada- A case study. PhD Thesis. University of Mysore (2002)
19. Kay. Nonconcatenative Finite State Morphology. EACL. pp. 2–10 (1985)
20. Aho, A.V., Sethi, R., Ulmann, J.D.: Compilers: Principles, Techniques and Tools. Addison wesley, Reading (1985)
21. Pal, U., Chaudhuri, B.B.: Indian script character recognition. Pattern Recognition 37, 1887–1899 (2004)
22. Cao, H.-L., Zhao, T.-J., Li, S., Sun, J., Zhang, C.-X.: Chinese POS tagging based on bilexical co-occurrences. Machine Learning and Cybernetics Conf. (2005)
23. http://www.indictrans.in
24. http://ccat.sas.upenn.edu/plc/kannada/