LANGUAGE IN INDIA

Strength for Today and Bright Hope for Tomorrow

Volume 14:6 June 2014
ISSN 1930-2940

Design of Derivational Morphological Analyzer for Kannada Language

Bhuvaneshwari C Melinamath, Ph.D. Scholar

==========

Abstract

We have developed a derivational morphological analyzer for Kannada Language. Derivational morphology deals with change of part of speech (POS) category from one basic category to another by the addition of derivational suffixes to the basic categories like noun, verbs and adjectives. Nouns can be derived from verbs and verbs can be derived from nouns and so on. There is no derivational morphological analyzer exists for Kannada. Existing systems have attempted only the inflectional morphology. The process of derivation is regular and productive in many instances for Kannada. But this is not true in all cases. Verbalizers are used in the process of derivation to verb. Nominalizers are used in derivation verb to noun. Finite state transducers are used for the implementation of the derivational analyzer. A set of verbs is

used to derive verbs from adjective. Another set of pronoun suffixes are used to derive nouns form adjectives. The accuracy of derivation analyzer is around 90% in the case of nouns and around 85% in the case of verbs.

Keywords— Part of Speech (POS), Natural Language Processing (NLP), Finite State transducers (FST).

1. Introduction

Natural Language Processing (NLP) is an area which is concerned with the computational aspects of the human language. The goal of the NLP is to analyze and understand natural languages used by humans and to encode linguistic knowledge into rules or other forms of representation. Statistical and machine learning algorithms have taken a lead over complex linguistic grammar. There has been a great progress in natural language processing, through the use of statistical methods trained on large corpora.

Morphology in the generative perspective started with the publication of Chomsky's remarks on nominalization (1970). Later on the foundation for generative morphology was laid by Mark Aronoff in his book 'Word formation in generative grammar'. According to Aronoff (1976) all regular word formation process are word based. A new word is formed or derived by applying a regular rule to a single already existing word. Both the new word and existing one are members of major lexical categories. Word formation rule operates on a single word, not on phrase or bound morpheme. Both input and output of WFR must be members of a major lexical category such as noun, verb, adjective, and adverb. Derivation on the other hand is a phenomenon which derives new words; derivation may be by adding an affix or by compounding. Words are formed from a combination of one or more free morphemes and zero or more bound morphemes. Free morpheme is units of meaning, which can stand on their own as words. Bound morphemes are also units of meaning; however, they cannot occur as words on their own: they can only occur in combination with free morphemes. From this definition, it follows that a word is either a single free morpheme, or a combination of a single free morpheme with other free and bound morphemes. The remaining part of the paper is organized as section2

deals with literature survey, section 3 deals with morphology and its types. Section 4 deals with results and section 5 deals with conclusion.

2. Literature Survey

In general there are several approaches attempted for developing morphological analyzer and Generator worldwide. Kimmo Koskenniemi (1983) has developed a two-level morphology formalism for Finnish language. Beesley (1996), has developed an Arabia finite state transducer for MA using Xerox finite state transducer (XFST). In case of Indian languages, (AU-KBC, 2010) the Research Centre of Anna University developed a finite state automata based morphological analyzer for Tamil language. Previous works show that very little work was done in computational aspect of Kannada: Prof. Kavi Narayan Murthy "Network and Process Model" handles only inflectional morphology; IIIT Hyderabad and Department of CALTS have done some work in respect of Kannada Morphology, but the approach is based on paradigm, i.e. suffix list based. Performance of the system is limited by the size of the dictionaries and derivational morphology is not handled. E. Roche and Y. Schabes (1997) have shown that Finite state technique is a vital tool in the implementation of natural language morphology.

From The literature survey, it is clear that derivational morphology of Kannada is not exploited to the extent it is necessary to handle this aspect, which otherwise hinders the NLP applications. In this paper an attempt is made to cover both the inflectional and derivational morphology for Kannada nouns and verbs. Morphology aspects are discussed in the next section.

3. Morphology

In linguistics, morphology is the study of the internal structure and the transformational processes of words. The internal structure of words is its morphemes. Each morpheme is an individual unit of meaning. Morphology is classified as inflection and derivational morphology.

3.1 Inflectional Morphology

This is the most significant part of morphology from the syntax point of view.

Inflectional morphology deals with words which are syntactically significant word forms projected towards morpho-syntactic functions, often overtly marked by corresponding bound morphemes. Inflection involves formation of a distinct word form of a given word, words are formed by the base or root words by the addition of certain affixes, to express certain grammatical relationships and functions. The changes are with respect to gender, number, case markers, personal suffixes in case of noun and pronoun, tense, aspect, and modal categories in case of verb. Inflectional categories vary with respect to language. For example, English does not have gender as an inflectional category, since its subject verb grammatical agreement rule does not require the gender information of the subject noun, but only person and number of the subject. In contrast, Kannada requires gender information along with person and number to mark the verb form to show the agreement with the subject. Inflectional words are words which are used in syntax to compose sentences, when different forms of the same lexeme are generated, and there is no change in category.

3.2 Derivational Morphology

Derivation is the process of formation of new words or inflect able stem from another word or stem. It typically occurs by the addition of an affix. The derived word is often of a different world class from the original. It may thus take on the inflectional affixes of the new word class. Derivation has the following characteristics:

- Typically produces a greater change of meaning from the original form.
- Is more likely to result in a form which has a somewhat idiosyncratic meaning.
- Often changes the grammatical category of a root
- Kindness is derived from kind.

Here are some kinds of derivational operations:

- Operations that change the grammatical category of a root. Verbs and adjectives can be turned into nouns: Consider the examples like amaze→ amazement, speak →speaker, perform→ performance, soft→ softness, warm → warmth
- Operations that change the valence (transitivity) of a root.

3.3 Kannada Derivational Morphology

Derivation morphology deals with change of category from one to another. Noun is derived from verb or adjective, as we have shown a sample of the addition of suffix. VERB + v saMdhi.

Nominalizers: nominalizers are used to derive nouns from verbs. Consider an example:

- Use of derivational suffix "vike"

maaDu 'do' (verb) + vike→ maaDuvike (noun) "process of doing".

- Causation Kannada suffix "isu"

prasne 'question'+ isu → prasnisu is verb which means questioning.

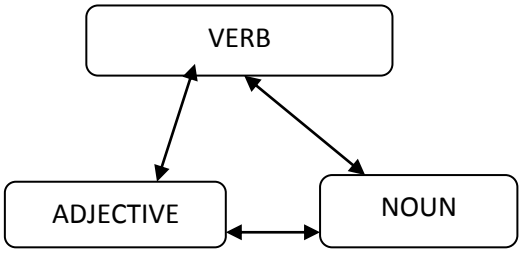

Figure 1: Derivation Morphology

Verbalizes A set of verbs used to derive verbs from nouns and adjectives. There is a set of verbs like 'aagu', biDu, iru, goLLu, aaDu, hoDe, maaDu which are used in deriving verbs from nouns.

Noun+verb=Verb. Consider the following examples.
gaaDi+hoDe= gaaDihoDe
nidde+maaDu=niddemaaDu

- Use of verb (aagu) 'become' in verbalization.

Most of the nouns in Kannada are derived as verbs by adding verb aagu to basic noun roots. The following examples illustrate the derivation Process.

- Noun to verb derivation using (aagu) verbalizer.

a) peTTu (Injury) 'noun' + aagu 'become/happen'→peTTaagu

maLe (rain) 'noun' +aagu→ maLeyaagu (raining)
beLe (crop)'noun' + aagu→ beLeyaagu (cropping)
paasu (pass) 'noun' +aagu→ paasaagu (paas)
ishTa (like) 'noun' + aagu→ ishTavaagu (like)

- Adjective to Verb Derivation Process

b) haaLu 'adjective'+ aagu →haaLaagu (get ruined)
suMdara 'beautiful' + aagu→suMdaravaagu (become beatiful)

- Use of Verb aagu in Complex verb Derivation

In Kannada we have a set of verbs which do not have aspect and tense inflections. Like beeku (want), beeDa (donot want), kuuDadu (no), bahudu etc. By adding verb 'aagu' to these verbs aspect and tense inflections can be added.

- Usage of aagu with Modal auxiliaries

c) beeku (want) 'verb' +aagu →beekaagu (wanted). Now beekaagu is a compound verb

- We had another set of conflicts like whether maataaDu 'talking'(maatu + aaDu is noun+verb) is compound should be stored in dictionary, or generated by morph system by adding verbalizes like aagu iru, aaDu, paDu, goLLu etc. finally We decided

to handle it through morph system.

Kannada language has complex morphology, many suffixes get added to form the complex inflected root. Consider the formation of verbal noun in example shown below:

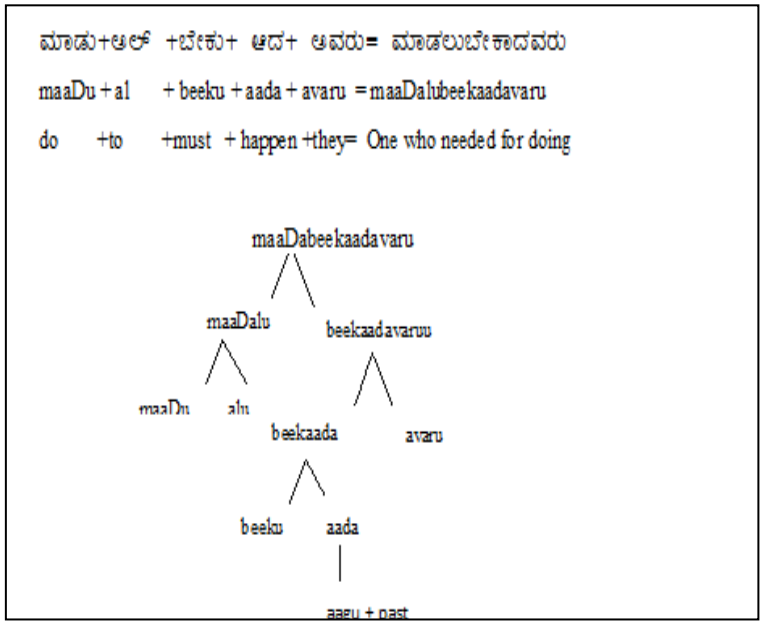

Figure 2. Sample Derivation Tree of Verbal Noun (Gerund)

3.4 Finite-state Transducers

Finite State Automaton (FSA) is a model of computation consisting of a finite set of states, a start state, an input alphabet, and a transition function that maps input symbol and current state to next state. A state transition usually has some rules associated with it that dictate when the transition may occur. The result given by a FSA is limited: either the string is accepted, or it is rejected. The transducer is defined as T = (Q, L, δ, qI, F,) where Q is a finite set of states, L a set of transition labels, qI Є Q the initial state, F ⊆ Q the set of final states, and δ: Q × L $\rightarrow 2^Q$ the transition function (where 2^Q represents the set of all finite sets of states). The set of transition labels is L = (Σ ∪ {ε} x (Γ ∪ {ε}) where Σ is the alphabet of input symbols, Γ the alphabet of output symbols, and ε represents the empty symbol. On next input symbol, it can move to state 2 without consuming any input symbols, and thus there is an ambiguity: is the system in state 1 or state 2, before consuming the letter a. Because of this ambiguity, it is more

convenient to talk of the set of possible states the system may be in. Thus, before consuming letter a, the NFA-epsilon may be in any one of the states out of the set {1, 2}. Equivalently, one may imagine that the NFA is in state 1 and 2 'at the same time': and this gives an informal hint of the power set construction 2^Q. The following table shows the transition table for Kannada derivational Morphology:

v-verb, n-noun, adj-adjective:

Transition State		Derivation Category	Derivation Suffix	
25-26		v>n	oosuga	
35-4		v>n	udu	
17-3		v>n	vike	
# Gerund = tense + udu, negative are like tense too:				
35	4	v>n	udu	gerund
# Nominalizers:				
17	3	v>n	vike	
# Relative Participles:				
# Adjective has no morphology as such. Derived adjectives behave				
# like nouns, morphologically.				
20	50	v>adj	(a)	rp
26	4	v>n	takkaddu→	compulsive_gerund
26	50	v>adj	takk(a) →	rp_compulsive
26	4	v>n	aaraddu	gerundn_capabilitative
26	50	v>adj	baarad(a)→	rp_prohibitive
30	50	v>adj	illada→	rp_negative
30	4	v>n	illaddu→	gerund_negative
35	4	v>n	baaraddu→	baaradu_gerund
35	4	v>n	lolladdu	olla_gerund
35	50	v>adj	loll(a)	rp_olla
35	50	v>adj	lollada	rp_ollada
50	94	adj>adv	aaga	particle_aaga
#50	94	adj>adv	aagalee	particle_aagalee
50	94	adj>adv	aagaloo	particle_aagaloo
50	94	adj>adv	aagina	particle_aagina
50	94	adj>adv	meele	particle_meele
50	94	adj>adv	naMtara	particle_naMtara
50	94	adj>adv	baLika	particle_baLika
50	94	adj>adv	oDane	particle_oDane
50	94	adj>adv	atta	particle_atta
50	94	adj>adv	haage	particle_haage
50	94	adj>adv	aMt(e)	particle_aMte

Transition State		Derivation	Derivation Suffix	
50	94	adj>v	aMtilla	existential_negative
50	94	adj>v	haagilla	existential_negative
50	4	adj>n	ashTu	particle_ashTu
50	3	adj>n	alli	particle_alli
50	0	adj>n	eDe	particle_eDe
52	94	adj>adv	aaga	particle_aaga
#52	94	adj>adv	aagalee	particle_aagalee
52	94	adj>adv	aagaloo	particle_aagaloo
52	94	adj>adv	aagina	particle_aagina
52	94	adj>adv	varege	particle_varege
52	94	adj>adv	tanaka	particle_tanaka
52	94	adj>adv	maTTige	particle_maTTige
52	94	adj>adv	atta	particle_atta
52	94	adj>adv	haage	particle_haage
52	94	adj>adv	aMt(e)	particle_aMte
52	100	adj>adv	vareginaparticle_varege	
52	94	adj>v	aMtilla	prohibitive
52	94	adj>v	haagilla	prohibitive
52	4	adj>n	ashTu	particle_ashTu
52	3	adj>n	alli	particle_alli
52	0	adj>n	eDe	particle_eDe
# Adj+Pron=Pron				
50	100	adj	aMtaha	similaritive
50	3	adj>n	aMtahadu	similaritive_pr_adu
50	3	adj>n	aMtahavu	similaritive_pr_avu
53	3	adj>n	avanu	
53	3	adj>n	avanobba(nu)	
53	3	adj>n	avanaata	
53	3	adj>n	avaniita(nu)	
53	3	adj>n	aata(nu)	
53	3	adj>n	aake(yu)	
53	3	adj>n	avaLu	
53	3	>n	avaLobbaLu	
53	3	>n	avaLaake	
53	3	adj>n	avaLiike(yu)	
53	100	>v	avaLillade	
53	3	adj>n	avaru	
53	3	adj>n	avarobbaru	
53	3	adj>n	avarellaru	
53	3	adj>n	avaraaru	
53	100	adj>v	avarillade	
53	3	adj>n	avugaLu	pr_avu
53	3	adj>n	ellavugaLu	pr_avu
100	0	n>v	ennu	+v_saMdhi
100	0	v	ennu	+v_saMdhi
100	0	n>v	uMTu	+v_saMdhi
100	0	n>v	aagu	+v_saMdhi

Box 1. Derivation Morphology Continued.

```
100  0       n>v     iru             +v_saMdhi

#100        0       adj>v   aagu             +v_saMdhi
#100        0       adj>v   iru              +v_saMdhi

100  0      adv>v   aagu            +v_saMdhi
100  0      adv>v   iru             +v_saMdhi

# External saMdhi with iru forms:
98   100    any     ide             +v_saMdhi_ide
98   100    any     ive +v_saMdhi_ive
98   100    any     iddu+v_saMdhi_iddu

# External saMdhi with sub_conj etc:

#98  100    any     aadudariMda+sub_conj
```

Box 1. Derivation Morphology Continued

3.5 Compounding

Compounding is a word formation process, whereby two or more words are combined to produce a new word. The meaning of the resulting word cannot be understood as the combination of meaning of combined words. Compounds yield new meaning, even though they are made of existing words, for example **greenhouse** is 'a building for plants', but not a house which is green in colour. In linguistics, word formation is the creation of a new word. Word formation is sometimes contrasted with semantic change, which is a change in a single word's meaning. The line between word formation and semantic change is sometimes a bit blurry; what one person views as a new use of an old word, another person might view as a new word derived from an old one and identical to it in form. Word formation can also be contrasted with the formation of idiomatic expressions, though sometimes words can form from multiword phrases.

Compounding involves two or more words rather than affixes. Syntactic rules are not involved in the internal structure of compound words. In syntax, morphologically complex words are not different from simple words. Internal structure of compounds is inaccessible to syntax.

Compound agrees with derivation in not allowing inflection inside. Hence compounds are syntactic atoms like other words. Processing of meaning of compounds from the constituents is not always easy: noun+noun compounding like redfort " keMpukoTe".

4. Experiments and Results

We have considered DoE CILL corpus for experimentation, a sample file Bank1.aci.out is considered, the size of the file is 1084 words. We observe the following derivation types in the corpus. More than 50 % words in the corpus are derivational types. This shows the importance of handling the derivational morphology.

Derivation Type	No of words
n>v	148
n>adj	115
Gerund	115
Compound	150
v>adj	80
v>adv	145

Box 1: Derivation Types in Bank1.aci.out

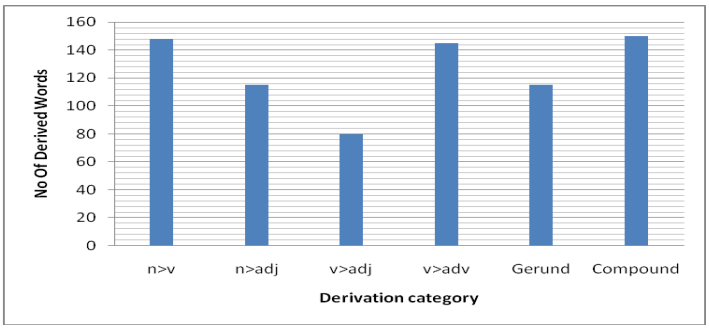

Figure 3. Derivation Category in Bank1.aci.out File

$$\text{Re}call = Tp/(Tp + fn) \qquad \ldots\ldots\ldots\ldots\ldots(1)$$

$$\text{Pr}ecision = TP/(TP + FP) \qquad \ldots\ldots\ldots\ldots..(2)$$

$$F - Measure = 2 * \text{Pr}ecision * \text{Re}call / \text{Pr}ecision + \text{Re}call \qquad \ldots\ldots(3)$$

Table 1. Showing Recognition Efficiency

Total Derivation Words	**842**
Correct Recognized	700
Wrong Recognized	53
Not Recognized	89
Recognized	753
Precision	95%
Recall	92%
Fmeasure	93%

5. Conclusion

We have attempted to develop a derivational morphological analyzer for Kannada, this tool is important in morphological process. Morphological analyzers and generators are useful in many NLP applications, like machine translation, parsing systems etc. Kannada language, having a complex morphology, designing analyzers or generator is a challenging job, due to agglutinative nature of Kannada.

References

http://www.iiit.net/ltrc/morph/morph_analyser.html.

Kavi Narayana Murthy(1999) : A network and process model for morphological analysis/ generation. Proc. Of the International Conference on South Asian Languages, Punjabi University, Patiala, India, 9-11 January.

Kenneth R Beesley (1983) : Arabic morphology using finite state operations. ph.D Thesis.

Koskenniemi (1983) : Two-level morphology, A general computational model for word form Recognition and production,. Master's thesis, Department of Genera Linguistics, Helsinki University.

Mark Aronoff. (1976) : Word Formation in Generative Grammar. MIT Press.

Noam, Chomsky. (1970) : Remarks on nominalization. In Roderick Jacobs and Peter Rosenbaum, eds., Readings in English transformational grammar, 184-221. Waltham, MA: Blaisdell.

Rao Uma Maheshwara Rao et. al. : CALTS. University of Hyderabad. On the description of morphological data for morphological analyzers and generators: A case of Telugu. Tamil and Kannada.

Sundaram Vijay et al.: Tamil morphological analyzer. "http://nrcfosshelpline.in/smedia/images /downloads/Tamil -opensource.odt

Y. Roche et. al (1997) : Introduction finite state language processing. MIT Press.

==========

Bhuvaneshwari C Melinamath Ph.D. Scholar
Department of Computer and Information Science
University of Hyderabad
Telengana
India
melinamathb@yahoo.com