LANGUAGE IN INDIA

Strength for Today and Bright Hope for Tomorrow

Volume 10 : 2 February 2010

ISSN 1930-2940

Automatic Nominal Morphological Recognizer and Analyzer for Sanskrit: Method and Implementation

Subhash Chandra, M.Phil., Ph.D. Candidate

Automatic Nominal Morphological Recognizer and Analyzer for Sanskrit: Method and Implementation

Subhash Chandra, M.Phil., Ph.D. Candidate

Abstract

The paper "Automatic Nominal Morphology Recognizer and Analyzer for Sanskrit: Method and Implementation" describes a system "Sanskrit Subanta Recognizer and Analyzer" developed for the degree of Master of Philosophy submitted to Special Centre for Sanskrit Studies (SCSS), Jawaharlal Nehru University (JNU) New Delhi .The system presents a model for Sanskrit nominal morphology (subanta) recognition and analysis (i.e. prakṛti-pratyaya vibhāga) for ordinary (laukika) Sanskrit texts. The authors while describing the components of this model also reported the research and development (R&D) done by author.
Some of the highlights of the developed system are as follows -

Keywords

Sanskrit Morphology, Sanskrit Noun Phrase Analyzer, Subanta Analyzer, Sanskrit Morphological System, Morphological Analysis Methods, Morphological Recognizer and Analyzer for Sanskrit, Sanskrit Noun Phrase, etc.

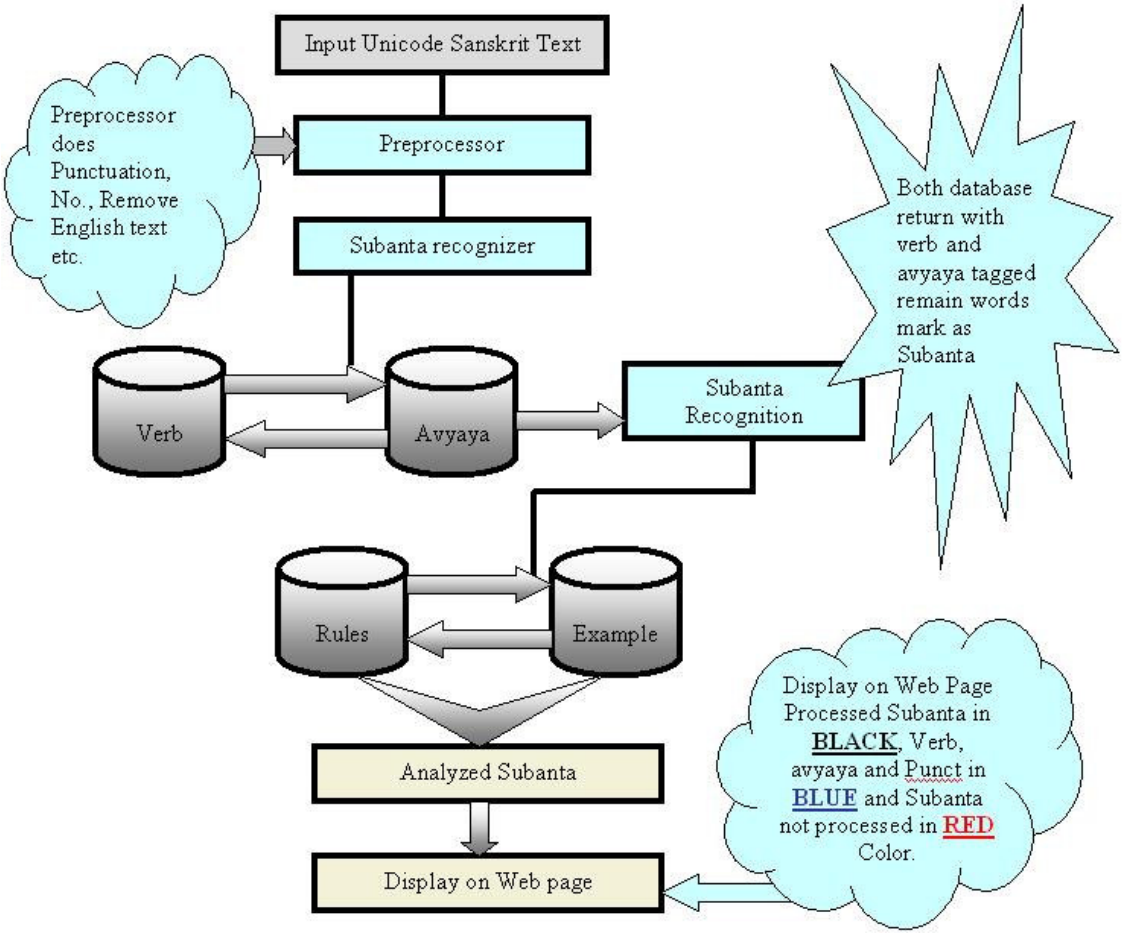
1. Introduction

Some of the highlights of the developed system are as follows –

- It a Nominal Morphological for Sanskrit.
- It is an online system available on http://sanskrit.jnu.ac.in/subanta/rsubanta.jsp. Therefore zero cost subanta analysis of Sanskrit text could be done by anyone anytime.
- Accept input in Unicode (UTF-8) Devnagari and Display in same format.
- It uses databases for Sanskrit subanta avyaya and verbs.
- It produced the vibhakti information as well as the subanta formulations of Pāṇini and later grammarians to parse a text for subanta.
- It is delivered in a web format using the OOP techniques in Java and SQL server.

- It can be used for M (A) T from Sanskrit to other languages.
- It can be used for self-reading and understanding of Sanskrit words.
- It is major part of Sanskrit Analysis tool.

The overall model of the developed system "Sanskrit Subanta Recognizer and Analyzer" is as follows-



2. Structure of Sanskrit nominal morphology

In a Sanskrit sentence, all non-verb categories are *subanta-padas,* which makes it essential to analyze these *padas* before any other computer processing can begin. Sanskrit *subanta* forms can be potentially very complex. They can include primary (*kṛdanta*) and secondary (*taddhitānta*),

feminine forms (*strīpratyayānta)* and compound nouns (*samāsa).* They can also include *upasargas* and *avyayas* etc. According to Pāṇini, there are 21 morphological suffixes (seven *vibhakitis* and three numbers 7 X 3 = 21), which can attach to the nominal bases (*prātipadika*) according to the syntactic category of the base, gender and end character of the base. Pāṇini has listed the *sup* suffixes *su, au, jas,am, au śas, ā, bhyām, bhis, ñe, bhyām, bhyas, ñas, bhyām, bhyas, ñasi, os, ām, ñI, os, sup.*

These suffixes are in the sets of these - (*su, au, jas*) (*am, au , śas*) (*ā, bhyām, bhis*) (*ñe, bhyām, bhyas*) (*ñas, bhyām, bhyas*) (*ñasi, os, ām*) (*ñi, os, sup*) for singular, dual and plural respectively. These suffixes are added to the *prātipadikas* (any meaningful form of a word, which is neither a root nor a suffix) to obtain inflected forms (*subanta padas*). *Prātipadikas* are of two types: primitive and derived.

The primitive bases are stored in g*aṇapāṭha* (collection of bases with similar forms) while the latter are formed by adding the derivational suffixes. They denote unity, duality and plurality respectively. Some words are only in the singular always, like *ekaḥ*(one), some are always dual like *dvi* (two), *akshi* (eyes) etc. and some are always plural like *apaḥ* (water), *dārāḥ* (wife) etc.

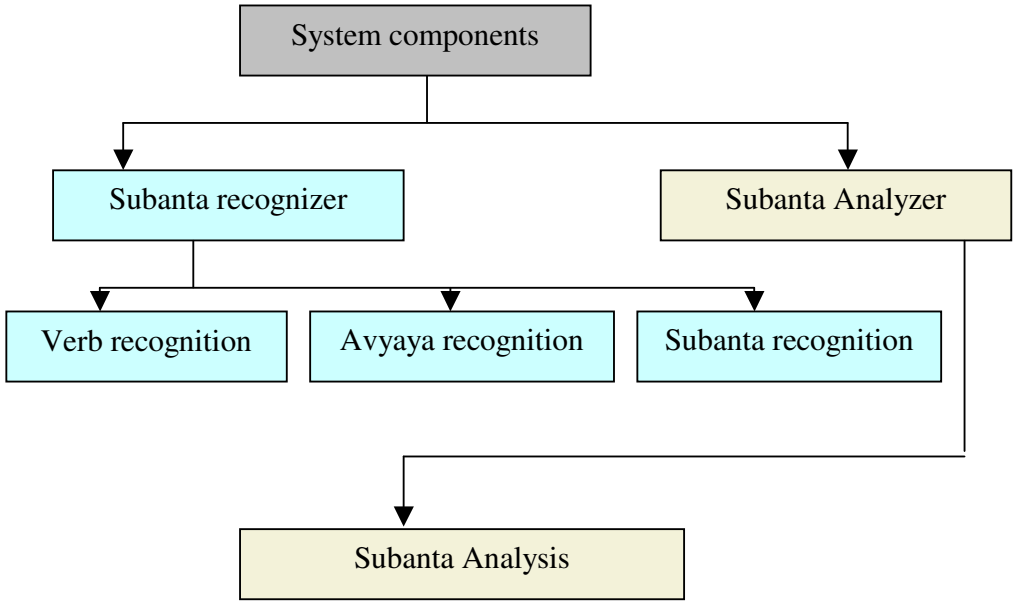
3. Previous work

Some work has been done by the Indian Heritage Group of the Centre for Development of Advanced Computing (C-DAC). The system called DESIKA' which claims to process all the words of Sanskrit, includes generation and analysis (parsing), has an exhaustive database based on *Amarakoṣa,* a rule-base using the grammar rules of *Pāṇini's Aṣṭādhyāyī* and heuristics based on *Nyāya & Mimāmsa śāstras* for semantic and contextual processing.

Huet developed a Grammatical Analyzer System which tags s*ubanata-padas* by analyzing s*andhi, samāsa and sup* affixation this system is available online at: http://pauillac.inria.fr/~huet/SKT/sanskrit.html. The Huet's system takes phrases and not full sentences or texts. The Special Centre for Sanskrit Studies, Jawaharlal Nehru University is currently engaged in the following research - *kāraka, verb* analysis, POS tagging of Sanskrit, online *Amarakoṣa*. Jha (2004) displayed a s*ubanta* generator built in Prolog. The RCILTS project under Prof. G.V. Singh at the School of Computer and Systems Sciences has prepared useful linguistic resources for Sanskrit.

4. System components

The work proposes the following modules as shown the tree diagram below –



4.1. *Subanta* recognizer

This module performs the following tasks in sequence – verb recognition, *avyaya* recognition and the *subanta* recognition.

4.1.1. Verb recognition

Sanskrit verb forms are very complex they carry tense, aspect, and number information all in the inflection forms. Sanskrit has about 2000 verb roots classified in 10 morphological and semantic classes. Further, these can have *ātmanepadī* and *parasmaipadī* forms in 10 *lakāra* and 3 x 3 persons and numbers combinations and can also be potentially. Mishra & Jha (2004) have done a rough calculation of all potential verb forms in Sanskrit to be around 10, 29, 60,000 plus. Storing all these verb forms would have been arduous. Therefore, we have using about 500 commonly used verbs and their forms. A sample listing follows -

dhātu_ id	**gaṇa**	**lat_pra_eka**	**lat_pra_dvi**	**lat_pra_bahu**
1	bhū	bhavati	bhavatah	bhavanti
2	edh	edhate	edhete	edhante
3	spardh	spardhate	spardhete	spardhante
4	gādhṛ	gādhate	gādhete	gādhante

Table-1

Basic verb root listing as in Pāṇini's *dhātupātha* [(organization of verb roots of first roots like bhv*ā*di (*bhuu, edh, aprdh* etc.)] has been done in the following format-

dhātu_ id	dhātu	gaṇa	Meaning
1	bhū	bhvādi	sattāyām
2	edh	bhvādi	vṛddhou
3	spardh	bhvādi	sañgharṣṅe
4	gādhṛ	bhvādi	pratiṣṣ ālipsayorgranthe cha
5	bādhṛ	bhvādi	viloḍane

Table-2

4.1.2 *avyaya* recognition

Sanskrit sentence must have a *tiṅanta-pada* and can have one or more *subanta-padas* (including *avyayas*). We have stored around 524 *avyayas* with Hindi meanings (for future use in M(A)T) in the following format-

Id.	avyaya	Meaning
1	a	ākśepa/sambodhana
2	akasmāt	achānaka
3	akānde	achānaka
4	aghoḥ	nikṛṣ /pāpī
5	aṅga	are/sambodhana

Table-3

4.1.3. *Subanta* recognition

After the verbs and avyayas have been identified, the remaining *padas* in the sentence are marked for *subanta* processing. Before the rule based reverse processing starts, the *padas* are checked in the exception list as given in the following format –

śabdarūpa	liṅga	prātipadika	pratyaya	Vibhakti/ vachana	rule_num
trayaḥ	PL	tri	jas	1.3	1.3.7, 7.3.109, 6.1.75, 8.2.66, 1.3.2, 1.4.109
trīn	PL	tri	śas	2.3	1.3.8, 7.3.109, 6.1.75, 8.2.66, 6.1.99
tribhiH	PL	tri	bhis	3.3	8.2.66, 1.3.2, 1.4.109
tirbhyaḥ	PL	tri	bhyas	4.3	8.2.66, 1.3.2, 1.4.109

Table-4

The rule based reverse processing will require the *gaṇa* information as stored in the following format -

Id	**śabda**
1	sarva
2	viśva
3	ubha
4	ubhaya
5	ḍa ara
6	ḍa ama

Table-5

4.2. *subanta* analyzer

Analysis of *subanta* is done according to the end-character of the forms. The present method stores all possible allomorphs of the 21 (7 x 3) *sup* suffixes in Sanskrit. The following table captures *subanta* dynamics of the sup suffixes. The examples given in table are for '*a*' ending masculine nouns-

vibakti	**Suffix**	**E**	**G**	**Value**
1	su	a	M	aḥ
1	au	a	M	au
1	jas	a	M	āḥ

Table-6 (1 = *prathamā*, E = Ending, G = Gender.)

Let us look at the following illustrations:

Sentence = *rāmaḥ gṛham gachchhan hasati*
(रामः गृहं गच्छन हसति।)

Ruled out padas = *hasatii* (recognized as verb)
(हसति)

Pada marked for *subanta* processing = *rāmaḥ, gṛham, gachchhan*
(रामः गृहं गच्छन)

Analysis:

rāmaḥ (रामः)
Base = *rāma (राम)*
ḥ → *su (सु)*
vibhakti = *su* (1-1) (सु)
Value of suffix = *aḥ* [*(su → s → ru → r → ḥ)* [P-4]

ः [(सु → स् →रु → र् → ः)]

rāma + *su* (1-1) (राम +सु)

1-1 *gachchhan* (*गच्छन्*)

Base = *gachchhat* (*गच्छत्*)

0→ *su* (*सु*)

vibhakti = *su* (1-1) (सु)

Value of suffix = *aḥ* [(*su* → *s* → *0*) [P-6]

ः [(सु →स् →0)]

Change in base = *gachchhat* →*gachchhan*

(*गच्छत* →*गच्छन्*)

gachchhat+ *su* (1-1) (गच्छत् +सु)

2-1 *gṛham* (*गृहम्*)

Base = *gṛha* (*गृह*)

am → *m* (*अम्* → *म्*)

vibhakti = *am* (2-1) (अम्)

Value of suffix = *a* (*am* →*m*)[P-7] अ (अम् →म्)

gṛha + *am* (2-1) (*गृह* +*अम्*)

5. The Tools and Technique Used

5.1 Frond End

Java **Server** Pages (JSP), HTML, Java Script

5.2 Java Object

A. Rsubanta (Accept form data and return processed data)

B. Preprocessor (preprocessed data, Subanta Recognition)

C. Sup_analyzer (Analyze subanta with the help of example and rule Database)

5.3 Back End

Database (SQL Server 2005) and text files in UTF-8

5.4 Web Server

Apache Tomcat

6. Limitations

The system has the following limitations -

- We are stored the commonly found verbs only. Though it is very unlikely that ordinary Sanskrit literature will overshoot this list, yet the system is likely to start processing a verb as *subanta* if not found in the database
- This work assumes initial *sandhi* processing, without which some results may turn out to be incorrect.

7. Problems and solutions

The R&D for this work so far has seen the following problems –

- Ambiguous *vibhaktis*
  - Same forms are available in the dual of nominative and accusative cases like- *rāmau,* dual of instrumental, dative and ablative cases like- *rāmābhyām,* plural of dative and ablative cases like- *rāmebhyḥ,* dual of relative and locative cases like - *rāmayayoḥ.* In neutar gender as well, the nominative and accusative singular forms may be identical as in *pustakam* (1-1 and 2-1). In such cases, [10] the system will give all possible results as in

    rāmau (रामौ) = au (औ) (1.2 & 2.2)
    rāmābhyām (रामाभ्याम्) =bhyām (भ्याम्)(3.2, 4.2 & 5.2)
    rāmebhyḥ (रामेभ्यः)= bhyas (भ्यस्) (4.2 & 5.2)
    rāmayayoḥ (रामयोः) = os (ओस्) (6.2 & 7.2)
    pustakam (पुस्तकम्) = su (सु) (1.1 & 2.1)
    hareḥ (हरेः) = ñas (ङस्) (5.1 & 6.1)

- Some *kṛdanta* forms (generally *lyap, tumun,* and *ktvā* suffix ending) look like *subanta* (for example - *vihasya vihāya, ādāya, gtvā, pathitvā* etc.). In such cases, the system may give wrong results like

    Vihasya (विहस्य) = viha (वि ंह)+ ñas (ङस्)[(6.1) (masculine '*a*' ending)]
    Gantum (गन्तुम्) = gantu (गन्तु)+ am(अम्) [(2.1) (masculine '*u*' ending)]
    vihāya (विहाय) = viha (विह)+ ñe(ै) [(4.1) (masculine '*a*' ending)]
    gtvā (गत्वा)= gtvā (गत्वा)+ su (सु) [(1.1) (feminine '*ā*'ending)]

To solve these problems, we are trying to store these *kṛdanta* forms of the 500 commonly found verb roots.

8. Results

System prints result in three color, Black, Red and blue. Black for processed subanta with analysis, Blue for Verb and Avyaya and Red for which word marked as subanta but system is not able to process. Here pasting a sample:

Input

वने एकः शशकः आसीत् । एकदा सः वृक्षस्य छायायां शयितः आसीत् । वृक्षात् एकं फलं तस्य मस्तके अपतत् । शशकस्य निद्रा भग्नाभवत् ।

Output

{ वने [वन् +ि चतुर्थी एकवचन] एकः [एक + सु प्रथमा एकवचन] शशकः [शशक (पुल्लिङ्ग) + सु , प्रथमा , एकवचन] [आसीत् _VERB] [। _PUNCT] [एकदा _AV] सः [तद् +सु , प्रथमा एकवचन] वृक्षस्य [वृक्ष + ◌स् , षष्ठी , एकवचन] छायायां [छाया +ि , सप्तमी , एकवचन] शयितः [शयित् +जस् /शस् /◌सि /◌स् , प्रथमा /द्वितीया , बहुवचन , पञ्चमी /षष्ठ ी, एकवचन] [आसीत् _VERB] [। _PUNCT] वृक्षात् [वृक्ष (पुल्लिङ्ग) + ◌सि , पञ्चमी , एकवचन] एकं [एक (पुल्लिङ्ग) + अम् , द्वितीया , एकवचन] फलं [फल (पुल्लिङ्ग) + अम् , द्वितीया , एकवचन] तस्य [तद् +◌स षष्ठी एकवचन] मस्तके [मस्तक +ि , सप्तमी , एकवचन] [अपतत् _VERB] [। _PUNCT] शशकस्य [शशक + ◌स् , षष्ठी , एकवचन] निद्रा [निद्र +टा , तृतीया , एकवचन] भग्नाभवत् _SUBANTA शशकस् य [शशक + ◌स् , षष्ठी , एकवचन] निद्रा [निद्र +टा , तृतीया , एकवचन] भग्नाभवत् _SUBANTA [। _PUNCT] }

Conclusion

In this paper, the authors have described a subanta analysis system and the intermediate results so far. The system has been delivered online in the Java servlet and relational database technology and is going to be very useful for processing of Sanskrit for any purpose. The system can be included as a very important component in any larger Sanskrit NL system by first identifying the s*ubanta- padas* in sentences and then splitting it into *prakṛti-pratyaya* according to *Pāṇinian* formulations. The system can be accessed online on *http://sanskrit.jnu.ac.in/subanta/rsubanta.jsp*

References

1. Grosz Barbara J., Jones Karen Sparck, Webber Bonnie Lynn (1986). *Readings in Natural Language Processing*, Morgan Kaufmann Publishers, Inc, California.

2. Huet's site http://pauillac.inria.fr/~huet/SKT/sanskrit.html (accessed on 1st October 2005).

3. Jha Girish N (1995). *Proposing a computational system for Nominal Inflectional Morphology in Sanskrit*, Proc. of national seminar on "Reorganization of Sanskrit Shastras with a view to prepare their computational database".

4. Jha Girish N (2003). *A Prolog Analyzer/Generator for Sanskrit Subanta Padas*, Language in India, volume 3.

5. Jha Girish N, (April 11-12th 1996). *Lexical conceptual structure and domain based machine translation system* (jointly with Prof. Kapil Kapoor, Prof. G.V. Singh,

presented at a symposium on Machine Aids for translation & communication, JNU, New Delhi.

6. Jha Girish N, March (2004). *Generating nominal inflectional morphology in Sanskrit*, SIMPLE 04, IIT-Kharagpur Lecture Compendium, Shyama Printing Works, Kharagpur, WB. Page no. 20-23.

7. Kapoor Kapil (11-12th April 1996). *Pāṇini's derivation system as a processing model* (to appear in the proc. Of "A Symposium on Machine Aids for Translation and Communication, School of Computer & Systems Sciences, J.N.U. New Delhi).

8. Kapoor Kapil (1985). *Semantic Structures and the Verb: a propositional analysis*, Intellectual Publications, New Delhi,

9. Mishra Sudhir K., Jha Girish N. (2004). *Identifying Verb Inflections in Sanskrit morphology*, in proc. of SIMPLE 04, IIT Kharagpur, Page no. 79-81.

10. Mishra Sudhir K., Jha Girish N. (2004). *Sanskrit kāraka analyzer for Machine Translation*, In SPLASH proc. of iSTRANS, Tata McGraw-Hill, New Delhi. Page no. 224-225.

11. Shastri, Bheemsen, *Laghusiddhantakaumudi (1st part)*, Bhaimee Prakashan, 537, Lajapatrai Market, New Delhi-06.

12. TDIL site, http://tdil.mit.gov.in/ (accessed on 20th February 2006).

13. Huet's site http://pauillac.inria.fr/~huet/SKT/sanskrit.html accessed on 20th February 2006).

14. Subash & Jha, Girish N. (2005). *Morphological analysis of nominal inflections in Sanskrit*. Presented at *Platinum Jubilee International Conference, L.S.I.* at Hyderabad University, Hyderabad, pp-34.

15. Subash and Jha Girish N. (2006). *Sanskrit Subanta recognizer and analyzer for machine translation*. in the *proceeding of 28th All India Conference of Linguists (28th AICL).* Department of Linguistics, Banaras Hindu University (BHU), India.

16. Subash. (2006). *Computational identification and analysis of complicated Sanskrit noun phrases proceeding of 2nd International Conference on cognitive Science (ICCS-2006).* Centre for Behavioural and Cognitive Sciences, University of Allahabad, Allahabad, U.P. India.

17. Subash. (2006). *Machine recognition and morphological analysis of Subanta-padas.* M.Phil dissertation submitted to J.N.U., New Delhi.

Subhash Chandra, M. Phil., Ph.D. Candidate
Scientific Officer, NLP Group
Centre for Development of Advanced Computing (CDAC)
E-2/1, Block-GP, Sector-5, Salt Lake Electronics Complex
Kolkata-700091
West Bengal, India
subhash.jnu@gmail.com
subash@cdackolkata.in