Computational Identification and Analysis of Sanskrit Verb-forms of bhvādigaṇa

Dissertation submitted to Jawaharlal Nehru University
In partial fulfillment of the requirements
for the award of the degree of

MASTER OF PHILOSOPHY

MUKTANAND AGRAWAL

SPECIAL CENTRE FOR SANSKIRT STUDIES
JAWAHARLAL NEHRU UNIVERSITY
NEW DELHI-110067
INDIA

2007

विशिष्ट संस्कृत अध्ययन केन्द्र
जवाहरलाल नेहरू विश्वविद्यालय
नई दिल्ली–११००६७

SPECIAL CENTRE FOR SANSKRIT STUDIES
JAWAHARLAL NEHRU UNIVERSITY
NEW DELHI-110067

July 30, 2007

DECLARATION

I declare that the dissertation entitled **"Computational Identification and Analysis of Sanskrit Verb-forms of bhvādigaṇa"** submitted by me for the award of the degree of **Master of Philosophy** is an original research work and has not been previously submitted for any other degree or diploma in any other institution/university.

(Muktanand Agrawal)

विशिष्ट संस्कृत अध्ययन केन्द्र
जवाहरलाल नेहरू विश्वविद्यालय
नई दिल्ली–११००६७

SPECIAL CENTRE FOR SANSKRIT STUDIES
JAWAHARLAL NEHRU UNIVERSITY
NEW DELHI-110067

July 30, 2007

C E R T I F I C A T E

This dissertation entitled "**Computational Identification and Analysis of Sanskrit Verb-forms of bhvādigaṇa**" submitted by **Muktanand Agrawal** to **Special Centre for Sanskrit Studies, Jawaharlal Nehru University, New Delhi-110067**, for the award of the degree of **Master of Philosophy**, is an original work and has not been submitted so far, in part or full, for any other degree or diploma of any University. This may be placed before the examiners for evaluation.

Dr. C. Upender Rao
(Chairperson)
Dr. C. Upender Rao
Chairperson
Special Centre for Sanskrit Studies
Jawaharlal Nehru University
New Delhi - 110067

Dr. Girish Nath Jha
(Supervisor)

Acknowledgement

I bow down to the almighty God and my Gurus who are the source of power and inspiration for me in every moment of my life. I am also immensely grateful to my parents who have made me whatever I am. For this dissertation, I am grateful to my guide Dr. Girish Nath Jha who not only initiated me in this unknown field of Computational Linguistics but also guided me with his constant encouragement and optimistic outlook. It is due to his efforts and motivation that I have been able to complete this work. I am grateful to Dr. Kapil Kapoor who has been directly and indirectly a great source of inspiration to me. I am also thankful to Dr. Shashi Prabha Kumar who always encouraged me in my studies. I wish to extend my thanks to all the other faculty members of the Centre for their guidance.

My friends and classmates have always been a source of confidence, energy and strength with their affectionate gestures. To say anything about their support will undermine it. I am grateful to all of them. Lastly, I want to thank my sister Shruti whose love and support is an invaluable treasure for me.

Contents

Transliteration scheme

अ	=	a
आ	=	ā
इ	=	i
ई	=	ī
उ	=	u
ऊ	=	ū
ऋ	=	ṛ
ॠ	=	ṝ
ऌ	=	ḷ
ए	=	e
ऐ	=	ai
ओ	=	o
औ	=	au
अं	=	aṁ
अः	=	ḥ
क्	=	k
ख्	=	kh
ग्	=	g
घ्	=	gh
ङ्	=	ṅ
च्	=	c
छ्	=	ch
ज्	=	j
झ्	=	jh
ञ्	=	ñ
ट्	=	ṭ
ठ्	=	ṭh
ड्	=	ḍ
ढ्	=	ḍh
ण्	=	ṇ
त्	=	t
थ्	=	th
द्	=	d
ध्	=	dh
न्	=	n
प्	=	p
फ्	=	ph
ब्	=	b
भ्	=	bh
म्	=	m
य्	=	y
र्	=	r
ल्	=	l
व्	=	v

श्	=	ś
ष्	=	ṣ
स्	=	s
ह	=	h
क्ष्	=	kṣ
त्र्	=	tr
ज्ञ्	=	j
ऽ	=	‘

Introduction

Morphological Analysis is an important area of NLP. A morphological analyzer takes a written or spoken text as input and analyzes it according to the rules of that language to provide the morphological analysis as output. Morphology is one of the constituting layers of linguistic structure of a language. It is concerned with the combination of morphemes to form new words. The importance of morphology and its analysis for a language vary according to the nature of that language. For most Indian languages, morphology is of much more importance because these languages contain most of the information in the words which show a rich inflectional and derivational tendency. That is why morphological analysis has been in the centre of NLP projects and activities in India. Various institutions are engaged in developing a variety of morphological tools such as analyzers, POS taggers, spell checkers, text processors and so on for Indian languages.

The present work is an attempt to make an analyzer for inflectional verb morphology of Sanskrit. Sanskrit acquires a unique place in Indian linguistic scenario. Apart from being one of the most ancient languages which also has a vast variety of literature, the language has scientifically defined structure. It has also left its impression on most of the languages of the subcontinent in morphology, syntax, grammar and many other linguistic aspects. Sanskrit follows a well defined process to derive and inflect words. The dominance of morphology in Sanskrit and other Indic languages leaves little scope for syntactic analysis. So, language almost follows free word-order and most of the information is contained in morphology. It is why morphological analysis has much more importance in developing any natural language understanding system for Sanskrit. The same is true for most of the other Indian languages.

The verb analyzer developed here recognizes verb forms in a given Sanskrit text. It further analyses these words to retrieve the related information. The core meaning of a Sanskrit verb is contained in the verb root or stem which becomes a base before verbal affixes are applied to it. The verbal suffixes which are called *tiṅ* terminations signify tense, aspect, mood, person and number etc. The analyzer follows the reverse

Pāṇinian approach to identify and analyze these constituent parts in Sanskrit verb forms. However, the present work has limited itself only to the first class of Sanskrit verb root list, i.e. *bhvādigaṇa*. It attempts to provide an analysis strategy which is able to identify the tiṅ affix in the verb root and thus recognize a tiṅanta verb in given Sanskrit text. Further it analyses the string a retrieve the base and prefix (if any) from a verb form.

The dissertation titled "Computational Identification and Analysis of Sanskrit Verb-forms of bhvādigaṇa" has following chapters –

- The first chapter "Computational Morphology and Sanskrit" introduces computational morphology with its various implementations in different systems. Computational morphology is mainly concerned with morphological analysis and generation. Morphological analyzers and generators use rules and lexicons of a language to accomplish respective tasks. Efforts for developing NLP tools for Indian languages have also focussed on computational morphology a brief survey of which is given. Also discussed are some prominent features of Sanskrit morphology, particularly verb morphology.

- The second chapter "Sanskrit Verb Morphology" discusses Sanskrit verb morphology in detail. Sanskrit verb forms are produced by conjugation of verb roots (*dhātus*) in ten *lakāras*. *Dhātus*, the meaning-bearer unit of verb, may be primitive (around 2000, listed in dhātupaṭha) or derived (from verb root or nominal stem). These verb roots, also called *dhātus*, are divided in 10 classes and roots of a single class follow the same process in most of the cases. The rules of Pāṇinian Aṣṭādhyāyī apply on these *dhātus* to form various verb forms in different paradigms from these verb roots. There are 10 *lakāras* which denote tense, aspect and mood in Sanskrit. There are 18 basic verbal terminations

- Third chapter "Morphological Analysis of Sanskrit Verb Forms" focuses on the analysis strategy that is to be followed in this work. Various approaches have been developed for morphological analysis. The popular ones are cut and

paste technique and finite state technique. The strategy followed in this work is based on reversed Pāṇinian approach. Pāṇinian system follows a step-by-step method to derive full-fledged verb form from a given root. The analysis methodology tries to sort out various morphemes in the verb forms

- The partial implementation of the verb analysis methodology to computer program is discussed in the fourth chapter "Verb Analyzer for Sanskrit." The analysis scheme is implemented to a computational program for identification and analysis of Sanskrit verb forms. The techniques used are Java programming language, Java Servlet Technology and JSP run on Apache Tomcat 4.0 Web-server. The system takes a running text of Sanskrit in Devanāgarī UTF-8 format as input. It first does the pre-processing and then tokenizes the text in order to separate all the words. Then it locates the *tiṅ* suffixes in tiṅanta forms and identifies them. In the next step, the system takes all the identified tiṅanta forms and separates the suffix in order to get the verbal base. In case, prefix is attached to it, it also searches for it.

The verb analyzer is a component of a larger system aimed at complete machine understanding of Sanskrit language. The system is available online at http://sanskrit.jnu.ac.in . The work done for this dissertation is a preliminary for complete verb analysis of Sanskrit and has immense scope of being developed as a comprehensive analyzer for Sanskrit verb forms.

Chapter-1

Computational Morphology and Sanskrit

1.0 Introduction

Morphology is a branch of linguistics that deals with word formation, analysis and generation. It is concerned with how words are formed or created in a language from smaller units in a systematic way. It has to find as well as to describe the mechanism behind this process. Each language has morphemes which are its smallest meaningful units. Realization of morphemes as part of a word is called morph. In formation of words from these smaller units, certain processes such as inflection, derivation and compounding etc. are involved. Also involved is morphotactics that determines how morphs should be put together to form words.

Computational Morphology (CM) is application of morphological rules in the field of computational linguistics. CL is an emerging area in Artificial Intelligence that is related to accomplish linguistic tasks through computational methods. CM deals with the processing of words in both their graphemic and phonemic forms. Its most basic task can be defined as taking a string of characters or phonemes as input and delivering an analysis as output. It has various practical applications from low-level applications to speech and language processing systems.[1]

Low-level applications

Low-level applications of CM are simpler tasks such as hyphenization, spelling correction, stemming etc.

[1] Harald Trost, Entry under "Morphology" in Oxford Handbook of Computational Linguistics

Hyphenation is almost exclusively done automatically. Segmenting words correctly into their morphs helps to solve the task. The major problem is spurious segmentations.

Spelling correction, in its simplest form, can be a mere comparison of the input against a list of word forms. But such a list will never contain all the words occurring in a text and enlarging the list will include more and more obscure words that will match with typos thus preventing their detection. Therefore, most systems use a root lexicon, plus a relatively small set of affixes and simple rules to cover morphotactics.

Stemmers are used in information retrieval to reduce as many related words and word forms as possible to a common canonical form which can then be used in the retrieval process. However, this canonical form is not necessarily the base form. The main requirement is – like in all the above tasks – robustness.

Another application is **text-segmentation** in Chinese, Japanese or Korean. In these languages words in a sentence are not separated by blanks or punctuation marks. Morphological analysis can be used to perform the task of word separation.

Natural language applications

Natural language processing (NLP) is a subfield of artificial intelligence and linguistics. It studies the problems of automated generation and understanding of natural human languages. Natural language generation systems convert information from computer databases into normal-sounding human language, and natural language understanding systems convert samples of human language into more formal representations that are easier for computer programs to manipulate.[2] CM has a wider scope of being applied in natural language processing systems involving parsing and/or generating natural language utterances in written or spoken form. Such applications can range from message

[2] *http://en.wikipedia.org/wiki/Natural_language_processing*

and information extraction to dialogue systems and machine translation. For many current applications, only inflectional morphology is considered.

In a parser, morphological analysis of words is an important prerequisite for syntactic analysis. Properties of a word the parser needs to know are its part-of-speech category and the morphosyntactic information encoded in the particular word form. Another important task is lemmatization, i.e., finding the corresponding dictionary form for a given input word, because for many applications a lemma lexicon is used to provide more detailed syntactic (e.g, valency) and semantic information for a deep analysis.

In generation, on the other hand, the task is to produce the correct word form from the base form plus the relevant set of morphosyntactic features.

Speech applications

A text-to-speech system takes (electronically stored) text as input and produces speech from it. Morphological analysis helps to solve two different tasks in such systems. One is to guide the grapheme-to-phoneme conversion. Characters are often ambiguous with respect to their translation into phonemes. Finding out the underlying morphological structure is necessary for solving the task correctly.

A less obvious application is the use of morphological analysis to help in determining the part-of-speech category of words. This is an important prerequisite of syntactic analysis which is the basis for coming up with a correct prosody.

Speech recognition is a field where morphological analysis can become ever more important. At the moment most available systems make use of full form lexicons and perform their analysis on a word basis. Increasing demands on the lexicon size on the one hand and the need to limit the necessary training time on the other hand will make morph-based recognition systems more attractive.

1.1 Morphological Analysis and Morphological Analyzers

Morphological analysis is the process in which a word is analyzed into its root word and associated morphemes. Morphological analyzers are programs used to provide morphological analysis of a language. They include recognition engine, lexicon/thesaurus, and algorithms to find out stem within an input word and identifying its suffixes.[3] They take as input a sentence, i.e. a sequence of words and retrieve related grammatical information such as lexical category, gender, number, person, tense etc. of every word. For example, for nouns they provide its gender, number and case information along with the root word. For verbs, information related to tense, aspect, mood etc. is retrieved. If a word has more than one meaning, the analyzer is supposed to return grammatical information for every meaning separately. [4]

It is the basic tool used in spell checker, grammar checker, parser and machine translation systems. To build a syntactic representation of the input sentence, a parser must map each word in the text to some canonical representation and recognize its morphological properties.

For the purpose of analysis of morphologically rich languages, the root and the morphemes of each word has to be identified. The combination of a surface form and its analysis as a canonical form and inflection is called a lemma. The main problems involved in the process are:

1. **Morphological alternations**: In a language the same morpheme may be realized in different ways depending on the context.

2. **Morphotactics:** Stems, affixes, and parts of compounds do not combine freely, a morphological analyzer needs to know what arrangements are valid.

---

[3] *http://www.acroterion.ca/Morphological_Analysis.html*

[4] Akshar Bharati, Vineet Chaitanya and Rajeev Sangal. "Natural Language Processing: A Paninian Perspective", Prentice-Hall of India, New Delhi, 1995.

There are many approaches for morphological analysis. A popular approach to morphological alternation is the cut-and-paste method in which the canonical form is derived by removing and adding letters to the end of a string. Another one is the finite state technology, which was introduced in early 1980s and is applied for automatic recognition and generation of word forms. It maintains that rules for morphological alternations can be implemented by finite-state transducers. It was also widely recognized that possible combinations of stems and affixes can be encoded as a finite-state network. An automaton containing inflected word forms can be upgraded to a morphological analyzer, for example, by adding a code to the end of the inflected form that triggers some predefined cut-and-paste operation to produce the lemma. Instead of cutting and pasting it at runtime, the entire lemma can be computed in advance and stored as a finite-state transducer whose arcs are labeled by a pair of forms. The number of nodes in this type of network is small, but the number of arc-label pairs is very large as there is one symbol for each morpheme-allomorph pair. A more optimal lexical transducer can be developed by constructing a finite-state network of lexical forms, augmented with inflectional tags, and composing it with a set of rule transducers. Lexical transducers can be constructed from descriptions containing any number of levels. This facilitates the description of phenomena that are difficult to describe within the constraints of the two-level model. Because lexical transducers are bidirectional, they are generally non-deterministic in both directions. If a system is only to be used for analysis, a simple finite-state network derived just for that purpose may be faster to operate. [5]

A morphological generator performs the reverse function of an analyzer. For every root and its features provided as input, it generates the word-forms in the given paradigms. Morphological analyzer and morphological generator are two essential and basic tools for building any language processing application for a natural language.

[5] *http://www.languageinindia.com/aug2006/parsingrajendran.pdf*

1.1.1 India's linguistic scenario

The Indian subcontinent is remarkably rich in languages. Linguistic diversity is a unique component of the cultural and political structure of this area. The 200 languages enumerated in the Census are a linguistic subtraction of over 1,600 mother tongues reported by the people indicating their perception of their linguistic identity and linguistic difference. [6] The linguistic diversity in India is marked by the fluidity of linguistic boundaries between dialect and language, between languages around state borders and between speech forms differentiated on cultural and political grounds. In spite of diversity, linguistic identity is thin because of the large size of population of the country. Some of the minor languages have more speakers than many European languages. The linguistic differences between the Indian languages, particularly at the grammatical and semantic levels, are less than expected, given their different historical origins. The mutual convergence of these languages due to their intensive and extensive contact have made India one linguistic area. Inter-translatability between those languages is therefore very high.

The languages of India can be historically categorized into four major language families:

Indo–European : This is the most prominent language family. Most of the major Indic languages leaving Dravidian languages belong to it. The Indo – European has the sub–families, Indo–Aryan and Dardic/Kashmiri. Major languages are Hindi, Sanskrit, Punjabi, Sindhi, Gujarati, Marathi, Bengali, and many other languages.

Dravidian: There are four prominent Dravidian languages – Telugu, Kannada, Tamil and Malayalam. All of them are spoken in the southern most part of the country.

Austro-Asiatic: It has Munda and Mon-Khmer/Khasi as sub-families.

Sino – Tibetan: Tibeto-Burman and Thai/Kempti are sub-families of Sino-Tibetan.

The Indo-European which is commonly called 'Indo-Aryan' has the largest number of speakers followed by Dravidian, Austro-Asiatic, also called 'Munda' and Sino-Tibetan which is commonly called the 'Tibeto-Burman'.

---

[6] *http://www.languageinindia.com/july2005/morphologynortheast1.html*

Andamanese: Andamanese is the fifth language family of India.[7] It comprises of the four language groups represented by the four speech communities of the Great Andamanese, the Jarawa, the Onge and the Sentinelese. Great Andamanese arguably is language family in itself.[8] All these languages are primitive languages and the speaking community of these languages are the primitive tribes who have been living in these islands for thousands of years, untouched from civilized society. They are people of the Negrito stock and it is not clearly known where they came here from.

1.1.2 Development of Language Technology in India

Technology Development in Indian Languages (TDIL)[9] was initiated by the Department of Information Technology, Ministry of ICT, Government of India with the objective of developing Information Processing Tools and Techniques to facilitate human-machine interaction without language barrier; creating and accessing multilingual knowledge resources; and integrating them to develop innovative user products and services. Since then, TDIL has been facilitating and supporting the development of language technology resources in all Indian languages, and promoting their dissemination. Towards this goal it has established 13 Resource Centers for Indian Language Technologies (RCILTS) in March 2000 at different Universities and Institutes.[10] These Resource centers are aimed at:

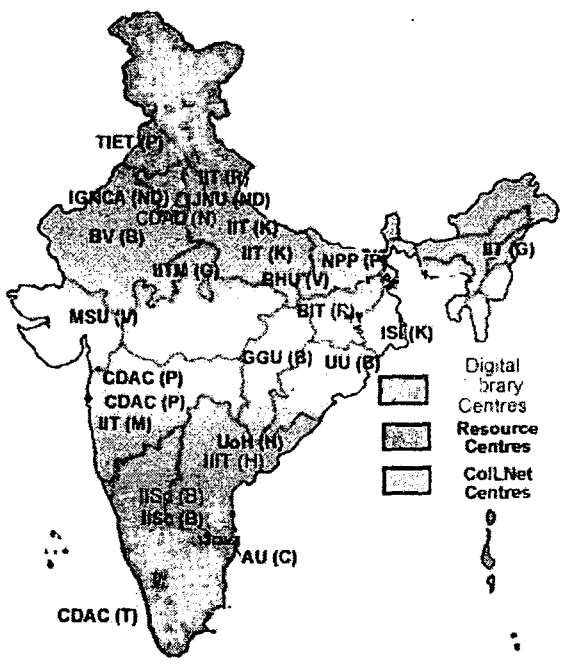
- Improving the quality of life of people of India by enabling to use Information Technology through Indian Languages..
- Developing new products and services for processing information in Indian Languages.
- Facilitating research in technology areas such as Machine Translation (MT), Optical Character Recognition (OCR), Text-to-Speech (TTS), and Speech Recognition in Indian Languages that leads to product development.

---

[7] Abbi, Anvita. 2006. Endangered Languages of Andaman Islands. Lincom Europa.
[8] Ibid.
[9] http://tdil.mit.gov.in/
[10] http://tdil.mit.gov.in/languagetechnologyresourcesapril03.pdf

List of Resource Centers

Resource Centre	Languages Associated With
Indian Institute of Technology, Kanpur	Hindi, Nepali
Indian Institute of Technology, Mumbai	Marathi, Konkani
Indian Institute of Technology, Guwahati	Assamese, Manipuri
Indian Institute of Science, Bangalore	Kannada, Sanskrit (Cognitive Models)
Indian Statistical Institute, Kolkata	Bengali
Jawaharlal Nehru University, New Delhi	Foreign Languages (Japanese, Chinese) & Sanskrit (Language Learning Systems)
University of Hyderabad, Hyderabad	Telugu
Anna University, Chennai	Tamil
MS University, Baroda	Gujarati
Utkal University,	Oriya
Department of Computer Science and Application	Punjabi
Thapar Institute of Engg. & Tech., Patiala	
ERDCI, Trivendrum	Malayalam
CDAC, Pune	Urdu, Sindhi, Kashmiri



A map showing main resource centers of TDIL across the country[11]

[11] Source: TDIL website, *http://tdil.mit.gov.in/*

A few other funding agencies are also supporting the language technology efforts. Several individuals in India and abroad and some industries have also been developing different language products/resources. An example of a morphological verb analyzer can be cited here that has been developed at JNU by a research scholar[12] for an extremely endangered language of Great Andamanese. As there are so many distributed efforts and with so many languages it is very difficult to keep track of the activities in this area. So, TDIL has tried to evolve an indexing system for the language technology resources, so that a suitable portal can be created to offer a variety of services to persons concerned with Indian language technologies. If all the resource centres follow this indexing and share their documents with TDIL, it would be easy to keep track of the progress and would be possible to encourage more people to participate in the development activities.

1.1.3 Survey of Morphological Tools for Indian languages

Morphological analyzer is an integral part of any Natural Language Processing system, especially in the context of Indian languages where morphology plays significant role due to high inflectional and derivational nature of these languages. For fixed word-order languages, the semantics of a word are primarily governed by its absolute and relative position within a sentence. But for free word-order languages, the position of words in the sentence cannot provide much clue about its semantics. As in the case of Indian languages, which are mostly free order, the semantics (part of speech and other subtleties) are heavily dependent on the surface realization of the word. Therefore, morphological analysis is inevitable to develop any NLP tool for Indian languages. Unlike English and various foreign languages, most of the Indian languages can be characterized by a rich system of inflections (*vibhakti*), derivation and compound formation. The numbers of word are derived from the root word by some specific orthographic rule in the Indian languages. A competent morphological analyzer is needed for a machine to deal with the lexicons of these languages.

[12] Choudhary, Narayan Kumar, 2006, "Developing a Computational Framework for the Verb Morphology of Great Andamanese", Unpublished dissertation at Centre for Linguistics, JNU.

The task of the morphological analyzer is to identify the structural components of a word, and glean information about it. While Indian languages in general are morphologically richer than languages like English, Dravidian languages are a lot more complex. Clearly, there is no way to list all forms of all words in a dictionary. Therefore, morphology is essential.

The development of Natural Language Processing in India has done significant progress since its beginning. A variety of tools have been developed for different purposes. These include machine translation systems, morphological analyzers and generators, POS taggers, spell checkers, text processors, word nets and many other tools. Given the morphological richness of these languages, numerous NLP tools related to morphology have been developed. The range covered is from major languages like Hindi, Telugu and Bengali to languages like Manipuri and Assamese. The survey of morphological analyzers, generators and other tools for morphological understanding of various Indian languages is given below. It mainly focuses on morphological analyzers and generators, taggers, spell-checkers and text-processors. However, efforts and tools of government institutions only have been covered leaving behind the initiatives of private sector which also is putting a lot of effort in this field.

1.1.3.1Morphological Analyzers for Indian languages:

Morphological Tagger for Hindi

Vasu Renganathan from University of Pennsylvania has developed a Hindi Morphological Tagger[13]. One has to enter a Hindi word or sentence and press 'Tag' to get tagged words as output. It is available in Unicode and ITRANS compatibility. The source code is written in PERL. The site also provides some of the data files

Morphological Analyzer for Telugu

[13] *http://ccat.sas.upenn.edu/plc/tamilweb/hindi.html*

Telugu is highly inflectional and agglutinative language. It exhibits features like vowel harmony, sandhi etc. Clitics, particles, vocatives are all part of the word. Auxiliary verbs are used in various combinations to indicate complex aspects.

RCILTS for Telugu, Univ. of Hyderabad claims to develop the only large scale morphological analyzer system built for Telugu.[14] It has been built, tested against corpora and refined over the past 10 years. The system uses a root word dictionary of 64,000 entries and a suffix list categorized into a number of paradigm classes. The basic methodology is to look for suffixes, remove them taking care of sandhi changes and then cross checking with the dictionary. Inflection, derivation, external sandhi are all handled. There is also a separate morphological generator that can put together the roots and affixes to construct complete word forms.

For POS Tagging to Telugu, a morphological analyzer developed by G. Uma Maheswara Rao from University of Hyderabad is being used. It gives the way to analyze each and every word in the Telugu corpus with the different possibilities.

Kannada Morphological Analyzer

RCILTS, Hyderabad has also developed a Kannada morphological analyzer and generator using their *Network and Process Model.*[15] A finite state network captures in a declarative and bidirectional fashion all the affixes, their ordering and the various permitted combinations. The process component takes care of sandhi changes when affixes are added or removed. This model makes it possible to develop a morphological analyzer, test it against a corpus and then get a generator of comparable performance with no extra effort since the same network is used both for analysis and generation. In this model, as it is claimed, a complete and detailed analysis is made at the level of each affix.

---

[14] TDIL magazine '*vishvabharata*' July 2003, accessible@ *tdil.mit.gov.in/Telugu-UOHJuly03.pdf*
[15] Ibid.

Morphological Analyzer and Synthesizer for Hindi [16]

Media Lab Asia Research Laboratory at IIT Kharagpur, [17] WB has developed a morphological analyzer for Hindi. It can identify the tense, aspect, modality, person, gender and number along with the root (*'dhaatu'*) of an inflected verb form. For this, the morphological analyzer uses a Directed Acyclic Morphological Structure. For nouns, the analyzer has to determine its *vibhakti* (inflection), suffixes and prefixes. It analyzes the lexical word groups corresponding to noun and determines the *karaka* (semantic role).

The group also aims to perform the decompositions for sandhi and samaasa (conjugating and compounding words) so as to have a powerful vocabulary for the system, and a generalized prefix and suffix handler. The DAM structure developed is based on the fact that possible inflections of a word can be identified by scanning the word backward from the end, one step at a time.

A morphological synthesizer has also been developed to generate the structural word given the root of the word, and the tense, aspect, mode, number, gender, and person. Hindi is a GNP language - i.e. the inflected forms of the verbs are mainly inflected using the gender, number and person information.

Morphological Analyzer and Synthesizer for Bangla [18]

Media Lab Asia Research Laboratory at IIT Kharagpur [19] has also designed a morphological analyzer and synthesizer for Bengali. The morphological analyzer can identify the tense, aspect, modality and person of an inflected verb form.

Bengali is a classifier language - where the inflection takes place mainly due to the tense, aspect and mode. For second and third persons, there is the concept of familiarity. The morphological analyzer has been modeled as finite state transducers which are given a

---

[16] *http://www.mla.iitkgp.ernet.in/technology.html*
[17] *http://www.mla.iitkgp.ernet.in/Resource/index.html*
[18] *http://www.mla.iitkgp.ernet.in/technology.html*
[19] *http://www.mla.iitkgp.ernet.in/Resource/index.html*

root and TAM and GNP information as input. The transducer's job is to generate the correct inflected form of the word by adding the appropriate suffix for Bengali.

Verbs are classified according to the vowels present in them. The modus operandi of the synthesizer is to determine the category of the root, given an input. It then accordingly synthesizes the ultimate form of the word depending on the input TAM GNP parameters. The number of exceptions is limited, and therefore easily traceable.

Oriya Morphological Analyzer (OMA)

PG Department of Computer Science and Application, Utkal University, Bhubaneswar has developed Oriya Morphological Analyzer (OMA), a computational model for the analysis and generation of Oriya language.[20] The major contents of OMA are:

i) Pronoun Morphology (PM),
ii) Inflectional Morphology (IM) and
iii) Derivational Morphology (DM).

The OMA has many constituent parts[21] such as OriNet Database (OD), which stores the Oriya lexicon (Only root words) whereas OMA Engine (OE) processes the system and Morphological Parsing (MP) parses the word according to orthographic rule. Decision Tree (DT) decides to classify the morphemes and different functional programmes by use of OMA.

The OMA system has been designed on the basis of Object-Oriented Approach (OOA) so that different functions can be easily added to or deleted from it. Pronoun Morphology and Inflection Morphology have been implemented in the OMA in such a manner that it successfully runs with the OriNet system, Oriya Spell Checker (OSC) and Oriya Grammar Checker (OGC). The OSC handles any type of word (derived, inflectional or root) using the OMA. It also provides sufficient interface for applications involved in Oriya Machine Translation (OMT), Word-Net for Oriya (OriNet), Oriya Spell Checker

[20] http://www.ilts-utkal.org/oriyamorphological.htm

(OSC) and Oriya Grammar Checker (OGC). All these developments have been worked out on the basis of the syntactic approach of Sanskrit language.

Developers have developed and implemented the Decision Tree (DT) and its respective algorithm of each type of morphology, through which OMA runs successfully. While performing morphological analysis, OMA not only deals with the study of words but also its morphemes. The OMA system is useful for various applications in developing NLP tools. For example, in OMT, there is need of root words, which is obtained through the OMA. Similarly, it has typical use in OriNet, the Word-Net for Oriya language, for searching any type of lexicon. Its Morphological Parser (MP) successfully handles ambiguous morphemes and provides different alternatives for them.

Morph Analyzer for Assamese and Manipuri[22]

RCILTS at Department of Computer Science & Engineering, Indian Institute of Technology Guwahati Assam[23] has developed morphological analyzers for Assamese and Manipuri. They are knowledge tools and work as an aid to MT system and Spell Checker.

Assamese Morphological Analyzer is based on the Stemming technique in which affixes are added/ deleted according to the linguistic rules. The derived words are verified with the existing Corpus/Dictionary to treat as valid words. This analyzer currently works with twenty linguistic rules, though more rules can be added without alteration in code. Modules are available in the form of API's for customization.

Manipuri Morphological Analyzer also uses stemming technique and does morphological processing with the help of the grammatical rules and the dictionaries of roots and affixes. A root dictionary containing 3000 root words and an affix dictionary containing 55 affixes are being used. Identification of linguistic rules for nouns and pronouns has also commenced. A model tagger is added to tag and analyze word. The

[21] http://www.ilts-utkal.org/publication/NLP/Abstract/abs-12.pdf
[22] *tdil.mit.gov.in/TDIL-OCT-2003/morph%20analyzer.pdf*

tagger tags the lexical category of the root and the grammatical category of the affixes. The code is written using Perl script and the user interface has been developed using Perl/Tk. The dictionaries are stored in MS-Access database.

Morphological Analysis in Tamil

Tamil is a verb final, relatively free-word order, morphologically rich and agglutinative language. Computationally, each root word in Tamil can take a few thousand inflected word-forms. Subject-verb argument is required for the grammaticality of a Tamil sentence. Tamil allows subject and object drop as well as verb less sentences. In addition, the subject of a sentence or a clause can be a possessive Noun Phrase (NP) or an NP in nominative or dative case. As Tamil is an agglutinative language, each root word can combine with multiple morphemes to generate word forms.

The following is the list of computational morphological analysis attempted and/or implemented for Tamil:[24]

1. Rajendran's Morphological Analyzer for Tamil: This was one of the very first efforts towards building a morphological analyzer for Tamil. It was initiated by *anusaraka* group of researchers under whose guidance Rajendran, Tamil University prepared this morphological analyzer for Tamil for Translating Tamil into Hindi at the word level.

2. AUKBC Morphological Parser for Tamil: [25] AUKBC NLP team under the supervision of Rajendran prepared a Morphological parser for Tamil. The API Processor of AUKBC makes use of the finite state machinery like PC Kimmo. It parses, but does notgenerate.

---

[23] *http://www.iitg.ernet.in/rcilts*
[24] "Parsing in Tamil: Present State of Art", S. Rajendran, Ph.D.
http://www.languageinindia.com/aug2006/parsingrajendran.pdf
[25] *http://www.au-kbc.org/frameresearch.html*

3. Vaishnavi's Morphological Generator and Analyzer for Tamil: Vaishanvi has built generators and analyzers for Tamil morphology. The generator implements the item and process model of linguistic description. It works by the synthesis method of PC Kimmo. The analyzer uses a hybrid model for Tamil. It is theoretically rooted in a blend of IA and IP models of morphology. It constitutes an in-built lexicon and involves a decomposition of words in terms of morphemes within the model to realize surface well-formed words-forms. Thus it tries to define a transformation depending on the morphemic nature of the word stem. The analysis involves a scanning of the string from the right to left periphery scanning each suffix at a time stripping it, and reconstructing the rest of the word with the aid of phonological and mopho-phonemic rules exemplified in each instance. This goes on till the string is exhausted. For the sake of comparison AMPLE and KIMMO models are implemented.

5. RCILTS-T's Morphological analyzer for Tamil:[26] Resource Centre for Indian Language Technological Solutions-Tamil, Anna University, Chennai has prepared a morphological analyzer for Tamil named as *Atcharam*. It takes a derived word as input and separates it into root word and associated morphemes. It uses a dictionary of 20,000 root words based on fifteen categories. It has two modules - noun and verb analyzer based on 125 rules. It uses heuristic rules to deal with ambiguities. It can handle verb and noun inflections. It is also used for extracting the root words from inflections in Tamil Search Engine and Online Tamil Dictionary. The system was developed using Java on Windows 2000 platform. It is expandable as more morphological rules can be added and Dictionary size can be increased. It has been tested with a file from CIIL Corpus producing approximately 75% result.

6. RCILTS-T's Morphological generator for Tamil:

Resource Centre for Indian Language Technological Solutions-Tamil has also prepared a morphological generator for Tamil. It is named as *Atchayam*. It generates words when Tamil morphs are given as input. It has two major modules – noun and verb generators. The noun section handles suffixes like plural markers, oblique form, case markers and

[26] *http://ns.annauniv.edu*

postpositions. The verb section takes tense and PNG makers, relative and verbal participle suffixes, and auxiliary verbs. It uses sandhi rules and125 morphological rules. It handles adjectives and adverbs. It has word and sentence generator interfaces.

Others

- **Ganesan's Morphological Analyzer for Tamil** to analyze CIIL corpus. It uses phonological and morphophonemic rules and takes into account morphotactic constraints of Tamil. An efficient morphological parser has also been built.
- **Kapilan's Morphological Analyzer** is especially built for analyzing Tamil Verbal Forms.
- **Deivasundaram's Morphological parser** built for a Tamil Word Processor. He too makes use of phonological and morphophonemic rules and morphotactics constraints of the language for developing his parser.
- **Ramasamy's Morphological Generator for Tamil:** Ramasamy has prepared a morphological generator for Tamil.
- **Winston Cruz's Parsing and Generation of Tamil Verbs** makes use of GS morph method for parsing Tamil verbs which does morphotactics by indexing. The algorithm simply looks up two files to see if the indices match or not. The processor generates as many forms as it parses and uses only two files.
- **Dhurai Pandi's Morphological Generator and Parsing Engine for Tamil Verb Forms:** It is a full-fledged morphological generator and a parsing engine on verb patterns in modern Tamil.

Punjabi Morphological Analyzer and Generator

The Advanced Centre for Technical Development of Punjabi Language, Literature and Culture, Punjabi University, Patiala has developed a Morphological Analyzer and Generator for Punjabi.[27] As generator, the software displays the list of all the possible word forms of any Punjabi root word, along with their respective grammatical information. As analyzer, it identifies the grammatical attributes of any Punjabi word and

[27] *http://www.advancedcentrepunjabi.org/punjabi_mor_ana.asp*

can also be used to search for any Punjabi word in it to know its root and other grammatical information.

The application areas of the software include, automatic spelling and grammar checking, natural language understanding, machine translation, speech recognition, speech synthesis etc part of speech tagging, parsing. The common man can also get in-depth information about the Punjabi words from the software. Knowing the grammatical information of a word helps in its proper and error free use in sentences. It can also help the beginners to learn new words and the specialists to create new terminology.

The database used in the software consists of more than 1.72 lakh Punjabi words, grouped into 22 word classes such as noun, personal pronoun, reflexive pronoun, verb, inflected and uninflected adverb, inflected and uninflected adjective, conjunction, interjection etc.

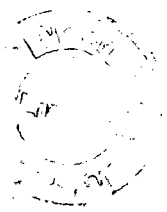
Morphological Analysers for multiple languages

Akshara Bharathi Group [28] at Indian Institute of Technology, Kanpur, India and University of Hyderabad, Hyderabad, India has developed morphological analyzers for different languages. These are available for online download for Hindi, Marathi, Telugu, Kannada and Punjabi. Downloaded system needs Linux Operating System, Perl, Perl enabled vim(only for Telugu), GDBM, Flex. The site claims that Telugu has 95% coverage (for arbitrary text in modern standard Telugu) and Hindi has 88% coverage.

IIT Bomb .y

RCILTS (CFILT - Marathi), IIT-Bombay, led by Dr. Pushpak Bhattacharya is a happening place for NLP in India. With an objective to enable Indian language information processing on internet, this project funded by MIT, Govt. of India has been working on creating Lexical Knowledge Bases like Wordnet , tools like MorphA and POS taggers for Hindi and Marathi and MTS with Universal Networking Language

[28] *http://ltrc.iiit.net/showfile.php?filename=onlineServices/morph/index.htm*



(UNL) as interlingua. Wordnet for Hindi and Marathi with lot of literature on it can be downloaded from the CFILT portal[29]. It also offers an overview of the UNL based Interlingua approach to MT with its online versions of English to Hindi MTS, Hindi to UNL conversion system, UNL to Hindi generation system. This centre is developing a POS tagger for Hindi using a data driven approach, making use of graphical algorithm Conditional Random Fields. This algorithm based tagger makes use of the spelling of the words, their lexical attributes and the suffixes to achieve an accuracy of about 90% on the BBC Hindi news corpora. It has a tagset of 30 tags with two pseudo tags S-START and S-END marking the sentence boundaries.[30]

1.1.3.2 Spell-Checkers for Indian languages

Tel-Spell: Spell Checker for Telugu

RCILTS-Telugu, Univ. of Hyderabad has developed a spell checker for Telugu named Tel-Spell.[31] The system has been tested on large scale corpora and enhancements and refinements have been going on for years. The new version is far simpler, more transparent, portable, well documented, and conforms to standards. Research work has also been taken up on developing stemming algorithms for Telugu. A pure corpus based statistical stemming algorithm has been developed. The performance of this stemmer for the spell checking application has been studied in various combinations with dictionary and morphology based approaches. A 10 Million word corpus developed has been used to build and test the performance. The performance of the system has been found to be satisfactory both in terms of detection and correction of spelling errors. The Telugu spell checker has also been integrated into AKSHARA advanced multilingual text processing system.

Bangla Spell-checker

---

[29] Resource Centre for Indian Language Technology Solutions (CFILT) - http://www.cfilt.iitb.ac.in/ (accessed: 14.10.2006)

[30] Srivastava, Manish et al., 2006, 'Conditional Random Field Based POS Tagger for Hindi', in the Proceedings of MSPIL, IIT-B, Mumbai

[31] *tdil.mit.gov.in/Telugu-UOHJuly03.pdf*

RCILTS-Bengali Indian Statistical Institute, Kolkata has developed a **Bangla Spell-Checker.**[32] In this spell-checker, only non-word errors are considered and not word errors like substitution, deletion, insertion, and transposition error. In Bangla, errors may occur due to phonetic similarity of characters or typographical mistakes. In this spell-checker, the main technique of error detection is based on matching the candidate string in the normal as well as in the *reversed dictionary.* Moreover, this approach is combined with a phonetic similarity key based approach where phonetically similar characters are mapped into a single symbol and a nearly-phonetic dictionary of words is formed. Using this dictionary, phonetic errors can be easily detected/ corrected. Here a candidate string first passes through the phonetic dictionary. If the word is not found in the dictionary and also failed to give suggestion then it tries to divide the word in root part and suffix part by separately verifying both. If an error is found, the spell-checker tries to provide suggestions. If it fails, it checks whether the string is a conjunct word generated by appending two noun words and suffix. Option for adding new words permanently or temporarily is provided in the spell checker.

The spell-checker uses several files containing root words and suffix words. The main dictionary contains about 60,000 root-words and 100,000 inflected words. Noun and verb suffix files are also used. The spell-checker works fast and almost correctly detects the non-word errors. However, it makes about 5% false alarm due to conjunct words formed by euphony and assimilation as well as proper nouns in the corpus.

Gujarati Spell Checker

RCILTS-Gujarati, Maharaja Sayajirao University, Baroda is developing a Gujarati Spell Checker.[33] It is based on a morphological analysis of Gujarati language. Morphological analyzer covers the analysis of Nouns and Verbs of the language and generates effective suggestions not greater then ten. All the non-Gujarati words along with different symbols are ignored and are not checked. The Gujarati Spell Checker is integrated along with Multi-Lingual Text Editor for the convenience to user. Currently the software is under

[32] *tdil.mit.gov.in/Bengali-ISIKolkataJuly03.pdf*
[33] *tdil.mit.gov.in/Gujarati-MSUniversityBarodaJuly03.pdf*

rigorous testing to find all the faults in the spell checker and morphological analyzer. Developers have plans to increase the size of the root dictionary covering the maximum possible words of the language and improving the algorithms for the correctness and efficiency of the spell checker. They also aim at building an independent spell checker which can be used on Unicode or ISCII compatible text.

NERPADAM - Malayalam Spell Checker

RCILTS-Malayalam, C-DAC, Thiruvananthapuram has designed a software subsystem named ***Nerpadam***[34] that can be integrated with Microsoft word as a macro or the Malayalam editor style-pad developed by them, to check the spelling of words in a Malayalam text file. While running as a macro in word, it functions as an offline spell checker and can be use with a previously typed text file only. Both off line and online checking are possible when it is integrated with the text editor. It generates suggestions for wrongly spelt words. The system adapts a rule cum dictionary-based approach for spell checking.

It incorporates a fully developed morphological analyser for Malayalam. This module splits the input word into root word, suffixes, post positions etc. and checks the validity of each using the rule database. Finally it checks the dictionary to find the root word. If anything goes wrong in this checking it is detected as an error and the error word is reprocessed to get 3 to 4 valid words, which are displayed as suggestion. The user can add new words into a personalized data base file, which can be added to the dictionary.

Tamil Spell Checker

RCILTS for Tamil at Anna University, Chennai has developed a spell checker for Tamil that takes care of the rich morphological structure of Tamil. After tokenizing the document into a list of words, each word is passed to the morphological analyzer which analyzes only the correct words. The morphological analyzer tries to split the suffix. In

[34] *tdil.mit.gov.in/Malayalam-CDACThiruvananthapuramJuly03.pdf*

case of error, it passes the word to spelling verification and correction phase to correct the mistake. When the correction of errors is completed, root word and all components are sent to morphological generator (for word forming), which then generate the possible corrected words as suggestions.

The Spell checker for Tamil helps the user to identify most of errors, which may occur while typing. The tasks implemented in Tamil Spell checker are Case marker, postposition checking and adjective checking for nouns, PNG marker checking for verbs, Adverb checking, and adjacent key error checking.

Spell-checker for Punjabi

RCILTS-Punjabi at Thapar Institute of Engineering & Technology, Patiala [35] has developed a spell-checker for Punjabi. Developers have retained multiple spellings for the commonly used words and have used Harkirat Singh's *Shabad Jor Kosh* as the base. Additional words were taken from the dictionaries published by Punjabi University and the State Language Department and the corpus developed by CIIL, Mysore. For generating the suggestion list, a study was conducted to discover the most common errors made by Punjabi typists. A list of similar sounding words and consonants was also complied and a suggestion list using this knowledge was generated based on reverse edit distance. In most cases, the right suggestion is presented as a default suggestion and the user just needs to confirm the default suggestion and proceed with the next error. Otherwise, the user needs to scroll through a list of suggestions and pick one as the right one. The spell checker supports text typed in any of the popular Punjabi fonts as well as ISCII encoded files. It is nearly complete now.

1.1.3.3 Text-processors for Indian languages

AKSHARA: An Advanced Multi-Lingual Text Processor

AKSHARA[36] is an advanced multi-lingual text processing tool. Various dictionaries, morphological analyzers, spell checkers, OCR systems, TTS systems; text processing tools have been included in it. Several text processing tools, Telugu spell checker and Telugu TTS have also been integrated. AKSHARA encodes texts in a standard character encoding scheme such as ISCII or UNICODE. Mapping to fonts is done only for the purposes of display and printing - all other operations are performed on character encodings. Attributes are included in an open XML style markup language called Extensible Document Definition Language (XDL) developed by us. This makes it easy to convert to and from various other encoding schemes thereby ensuring highest levels of portability and platform independence.

A unique feature of AKSHARA is that it understands the script grammar and warns if one tries to build ungrammatical syllables. AKSHARA has been successfully used to clean up all the corpora at CIIL, Mysore.

AKSHARA is platform independent and also claimed to be also robust and reliable. AKSHARA has been successfully used to develop a 10 Million word corpus of Telugu. Full support for Regular Expressions and Finite State Machines is being integrated. AKSHARA is unique in providing multilingual email sending as well as receiving facilities. AKSHARA also enables one to develop interactive web pages in Indian languages and English. These web pages work across platforms and browsers. All this is made possible through WILIO technology. AKSHARA is freely available.

Bilingual Punjabi/English Word Processor:

A bilingual Punjabi/English word processor named **Likhari**[37] has been developed by RCILTS-Punjabi at Thapar Institute of Engineering & Technology, Patiala. **Likhari** supports word processing under the windows environment and allows typing and

---

[35] *http://tdil.mit.gov.in/Punjabi-TIETPatialaJuly03.pdf*

[36] *http://ildc.gov.in/telugu/htm/Akshara.htm*

processing in Punjabi Language through the common typewriter keyboard layout. It has MS-Word compatible features and commands. It provides a number of features that make the use of Punjabi Language on a computer easy and provides a number of tools to increase the efficiency of the user. These tools include Bilingual Spell Checker with suggestion list, on-screen keyboard layouts with composition reference for Punjabi language typing, bilingual search and replace, sorting as per the language, alphabetical order, technical glossaries and onscreen Bilingual dictionaries.

The main features of **Likhari** are:

- Very simple user interface
- Online active Keyboard for users who do not know how to type in Punjabi.
- Choice of Phonetic, Remington and Alphabetic Keyboard layouts with composition reference.
- Bilingual Spell Checker for Punjabi and English.
- Bilingual Search and Replace.
- Support for sorting the text in English or Punjabi as per alphabetical order.
- Extensive help at various levels to make it easy for the user to learn.

Nashir – Word processor for Urdu and other languages:

RCILTS-Urdu, Sindhi & Kashmiri, Centre for Development of Advanced Computing, Pune has developed a word-processor named ***Nashir.***[38] It is said to be capable of creating documents in Perso-Arabic languages, and at the same time to layout complete newspapers and magazines in Urdu, Sindhi, Kashmiri, Arabic and Persian. Each document of the Nashir consists of a number of pages on which text blocks, graphics, etc. can be placed. Nashir is also well suited for publishing segment. Spellchecker support which uses a base-dictionary is added in the full version.

Nashir supports Nastaliq True Type fonts (presently 2 fonts) and Naskh fonts (presently 12 fonts) along with fonts for Sindhi and Kashmiri. It supports C-DAC & Phonetic Keyboards as well as user-defined keyboards. It provides various drawing objects and

---

[37] *http://tdil.mit.gov.in/Punjabi-TIETPatialaJuly03.pdf*

[38] *tdil.mit.gov.in/UrduSindhiKashmiri-CDACPuneJuly03.pdf*

also supports the OLE Automation. *Nashir* supports Urdu, Sindhi & Kashmiri along with English. A transliteration engine (uTRANS) has also been implemented which converts an ".aci" (ISCII file) into transliterated version in Urdu script (Naskh/ Nastaliq). Rule based transliteration has been developed for Hindi & Punjabi. The user can save the document as HTML page, and thus Naskh as well Nastaliq scripts can be viewed on the Internet.

Phonetic keyboard is supported. Both Horizontal and Vertical kerning is provided to manually adjust a text. It has Horizontal and Vertical rulers in the GUI, dynamic font settings for the Urdu and English fonts.

1.2 Sanskrit Morphology

1. 2.0 Sanskrit morphology

Sanskrit has two-fold morphology- nominal and verbal. In Sanskrit, a syntactic unit is called *pada.* A Sanskrit sentence, according to **Cordona[39] (1988),** can be represented in following formula

$$(N\text{-}E^{n})p\ldots(V\text{-}E^{v})p.$$

Pada can be nominal (*subanta*) or verbal (*tiṅanta*). These forms are produced by inflecting the stems and hence they are part of Sanskrit inflectional morphology. The derivational morphology in Sanskrit includes formation of derived nominal and verbal stems. Both of them can be formed by either nominal stems or verb roots.

1. 2.1 Inflection

Sanskrit is very rich in inflections. The inflection morphology involves formation of two kinds of words or *padas*: *subanta padas* (nominal words) and *tiṅanta padas* (verb forms). The Pāṇinian analysis for Sanskrit has categorized each and every usable word under these two categories. A word cannot be used in the language unless it is one of them. Even the indeclinable words (*avyayas*) are first assigned to the first category, and later the characteristics of it are deleted. The two categories of *padas* are named so according

to the set of affixes which is affixed to them. These are called sup suffixes and tiṅ suffixes respectively, thus creating *sup+anta* = *subanta* and *tiṅ+anta* = *tiṅanta padas*.

Nominal inflection

Nominal inflection in Sanskrit includes all non-verb categories, i.e. *subanta-padas*. 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 called sup suffixes to be attached to the nominal bases (*prātipadika*).

Pāṇini has listed 21 *sup* suffixes:[40]

su, au, jas, am, auṭ, śas, ṭā, bhyām, bhis, ṅe, bhyām, bhyas, ṅasi, bhyām, bhyas, ṅas, os, ām, ṅi, os, sup.

These suffixes are grouped in seven sets of three affix each- (***su, au, jas***) (***am, auṭ, śas***) (***ṭā, bhyām, bhis***) (***ṅe, bhyām, bhyas***) (***ṅasi, bhyām, bhyas***) (***ṅas, os, ām***) (***ṅi, os, sup***).[41] The 7 sets basically indicate seven cases and the *vibhakti*-s used for their denotation. The three in each set are meant to indicate three numbers singular, dual and plural[42] respectively. These suffixes are added to the *prātipadikas*[43] (any meaningful form of a word, which is neither a root nor a suffix) to obtain inflected forms (*subanta padas*). The further modifications of the sup affixes appended to the *prātipadika* depends mainly on the gender and end character of the base.

prātipadikas are of two types: primitive and derived. The primitive bases are stored in *gaṇapāṭha* (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), *akṣi* (eyes) etc. and some are always plural like *apaḥ* (water), *dārāḥ* (wife) etc.

---

[39] **George Cardona, 1988** Pāṇini, His Work and its Traditions, vol. i (Delhi: **MLBD, 1988)**

[40] svaujasamauṭchaṣṭābhyāmbhisṅebhyāmbhyasṅasibhyāmbhyasṅasosāmṅyossup

[41] supaḥ

[42] dvyekayordvivacanaikavacane

[43] arthavadadhāturapratyayaḥ prātipadikam 1/2/45; kṛttaddhitasamāsāśca 1/2/46

Verbal inflection

The *tiṅ* terminations are 18 in number. They are added to the verb root which can be primitive or derived. The *tiṅ* terminations are divided in 2 equal sets in two *padas: parasmai* and *ātmane.* The terminations in each set are again viewed as 3x3, i.e combination of three persons (first, second and third) with three numbers.

1.2.2 Derivation

Sanskrit is rich in derivation. Both *subanta padas* and *tiṅanta padas*, before they are inflected, generally undergo certain derivational operations. Derivation includes suffixes which are added to both nominal stems and verbal roots to form new words, again nominal or verbal. In other words, newly derived nominal words or verb roots can be formed by nominal stems as well as from verbal stems. The suffixes are well categorized with their meanings and environments.

Nominal Derivatives:

Nominal derivatives are the words formed by adding derivational suffixes to nominal words. These new words can be nominal stems as well as verbal stems. The nominal stems which are derived from the nominal stems can be mainly categorized under taddhitāntas and strīpratyayāntas. The nouns derived from verb roots are called *kṛdantas.* These are derived by affixing kṛt suffixes to the verb roots.

The nominal forms formed from other nominal stems are called *taddhitāntas*- derived by adding taddhita suffixes to nominal stems. The secondary derivative affixes are called *taddhita,* which derive secondary nouns from *prātipadikas.* For example - *dāśarathī, gauṇa* etc. Pāṇini has listed many *taddhita* suffixes some of which are- *a, akañc, ac, añ, aṇ, at, iṣṭhan, īyasun, kan, ḍhak, ḍhañ, tamap, tarap, tayap, tal, tyap, tral, dvayasac, fak, matup, mātrac, yak, yat, yañ, ḍāc, kha, gha, cha, uraca, ṭhak, ṭhañ, ṭhan, na, ha, va, vatup* etc. For example, *dākṣī, kva, aśvakaḥ, viśvajanīnam, kṣatriyaḥ, mālīyaḥ, raivatikaḥ, dāṇḍikaḥ, laghutamaḥ, gurutaraḥ, gārgyāyaṇaḥ, iha, balavān* etc.

Sanskrit also has eight feminine suffixes *ṭāp, cāp ḍāp, ṅīṣ, ṅīn, ṅīp, uṅ* and *ti* etc. and the words ending in these suffixes are called *strīpratyayānta.* For example - *ajā, gaurī, mūṣikā, indrāṇī, gopī, aṣṭādhyāyī, kurucarī, yuvatī, karabhorū* etc.

The nominal morphology of Sanskrit distinguishes nouns and adjectives on the one hand and pronouns on the other. There is no absolute distinction between nouns and adjectives with respect to inflection, but modifiers take the number and gender of the terms they qualify.

Pronouns are personal (first and second person), demonstrative (deictic), interrogative and relative. Pronouns other than personal pronouns observe gender distinctions. There are also endings particular to pronouns.

Sanskrit also has indeclinable terms, including particles such as the connective *ca* 'and' (sentence and nominal connective) and the negative particle *na*, as well as terms such as *yad, yadi* 'if', *tad, tarhi* 'then', *karhi* 'when?', *tataḥ* thence, *yataḥ* whence, *kutaḥ* whence?, *yadā* when, *tadā* then, kadaa when?, yatra where, tatra there, kutra where?, derived from pronominals.

There are also preverbs, which regularly occur immediately preceding a verb or another preverb – although in vedic they can be separated from a verb or follow it- as well as pre- and post-positional terms like adhi, anu, which co-occur with particular case-forms.

Indeclinable NPs

avyaya subanta-padas remain unchanged under all morphological conditions [44]. According to Pāṇini [2.2.82][45], affixes *cāp, ṭāp, ḍāp,* (feminine suffixes) and *sup* are deleted by *luk* when they occur after an *avyaya.* Pāṇini gives definitions for identifying certain word forms as avyayas which also include compound forms.

---

[44] sadṛṁ triṣu liṅgeṣu sarvāsu ca vibhaktiṣu |
vacaneṣu ca sarveṣu yanna vyeti tadavyayam || - gopatha brāhmaṇa

[45] avyayādāpsupaḥ 2/4/82

Verbal Derivatives:

Verbal derivatives may be nominal stems or verbal stems derived from verb roots or stems. The nouns that are derived immediately from verbs comprehend a great variety of terms. There are two principal classes of their classification:

- Adjectives (Attributives)
- Names (Substantives)

These different nouns are formed by affixing certain terminations to the verb root, which is modified in a greater or lesser degree, and then forms the inflective base. These terminations are numerous. The greater number has a very limited application while others comprehend a large class of words. The primary derivatives are called *kṛdanta.* The primary affixes are to be added to verbs to derive substantives, adjectives or indeclinable *kṛt.* For example *paṭhitavyam, pātavya, paṭhanīya, pacelima, jeyam, deyam, karttā, kumbhakāraḥ, janamejayaḥ, pāṭhakaḥ, paṭhantī, gantum, khāditum, svapnam gatiḥ, gatvā, vihāya, ādāya* etc.

Verbal derivatives also include participles, infinitives, gerund etc.

1.2.3 Compounding

Simple words (*padas*), whether substantives, adjectives, verbs or indeclinables, when added with another *subanta pada* form *samāsa* (compound). Sanskrit *samāsas* are divided into four categories, some of which are divided into sub-categories. The four main categories of compounds are as follows:

(1) Adverbial or *avyayībhāva,*

(2) Determinative or *tatpuruṣa,*

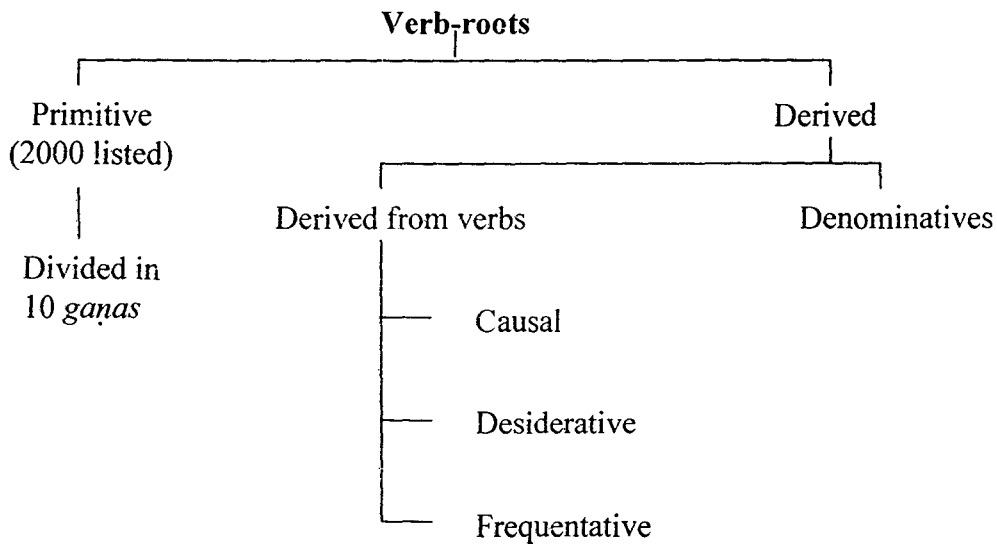
(3) Attributive or *bahuvrīhi* and

(4) Copulative or *dvandva.*

dvandva and *tatpuruṣa* compounds may be divided into sub-categories also. tatpuruṣa has dvigu and karmadhāraya as its sub-categories.

1.2.4 Sanskrit Verb-Morphology:

The verb forms are derived from verb-roots or *dhātus*. These *dhātus* are encoded with the core meaning of the verb. These can be primitive[46] or derived[47]. Primitive verb-roots, which are around 2000 in number, have been listed in a lexicon named *'dhātupāṭha'*. They are divided in 10 groups/classes called *gaṇas*. All the verb-roots of a group undergo somewhat similar inflectional process. Derived verb-roots may be derived from primitive verb-roots or from nominal forms. Prefixes also play an important role as they can change the meaning of a verb root. These roots then have to undergo various inflectional suffixes that represent different paradigms. In this process, the base or root also gets changed. The chart given on the next page gives an overview of Sanskrit verb roots.



Derived Verb-roots:

Derived from verb-roots:

1. **Causals *(nijanta)-*** Causals are formed by adding *'ṇic'* suffix to a primitive verb root. They convey the sense that a person or thing causes another person or thing to perform the action or to undergo the state denoted by root.

[46] *bhūvādayo dhātavaḥ* (*Pāṇini* 1/3/1)

2. **Desideratives *(sannanta)-*** Desiderative of a primitive verb root is formed by adding *'san'* affix to it. It conveys the sense that a person or thing wishes to perform the action or is about to undergo the state indicated by the desiderative form. Any basic verb-root or its causal base may have a desiderative form.

3. **Frequentatives *(yaṅanta)-*** Frequentatives verbs import repetition or intensity of the action or state expressed by the root from which it is derived. Can be of two types:

   a. *Ātmanepada* Frequentative *(yaṅanta)* – *'yaṅ'* affix added
   b. *Parasmaipada* Frequentative (*yaṅluganta*) – '*yaṅ*' is added but deleted

An illustration is given below of formation of derived verb-roots from a primitive verb root *bhū*.

bhū (to be)----------+--- (+*ṇic*) ---- *bhāvay* (to cause someone or something to be)
                         +--- (+ *san*) ---- *bubhūṣ* (to desire to be)
                         +--- (+ yaṅ) ---- *bobhūy* (to be repeatedly)
                                           (yaṅ deleted) +--- *bobho/bobhav*

These derived verb-roots, however, undergo similar operations, with some specifications, to form verb forms.

Roots derived from nominal words:

A large number of Sanskrit verb-forms can be derived from nominal words. These are known as 'nāmadhātus' (denominatives). Taking a nominal word as head, various derivational suffixes are added to these to form nominal verb-roots. The sense conveyed by the nominal verb root depends upon the suffix added to it. Yet, denominatives commonly import that a person or thing behaves or looks upon or wishes for or resembles

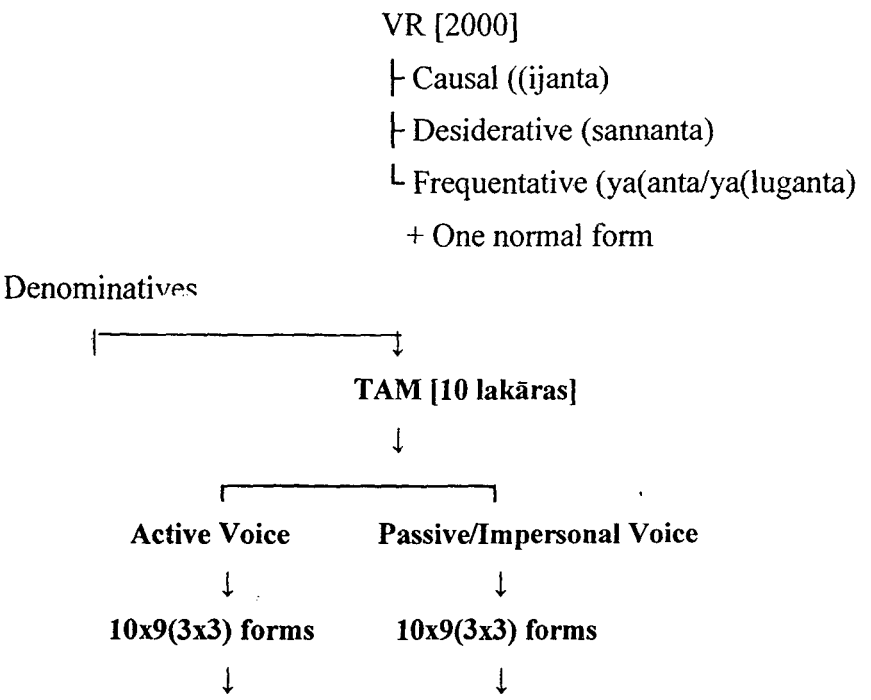
---

[47] *sanādyantā dhātavaḥ* (*Pāṇini* 3/1/32)

a person or thing denoted by the noun. These denominatives, however, can be innumerable as there is no end to nominal words in Sanskrit.

Process of formation of Sanskrit verb forms:

A Sanskrit verb root may take various forms. There are ten lakāras that represent Tense, Aspect and Mood. Inflectional terminations are 18 in number. These are divided in two groups – Parasmaipada and Ātmanepada, each having 9 affixes which is a combination of 3 persons x 3 numbers. A verb is conjugated in either pada, though some of the roots are conjugated in both. For each different lakāra, a root is affixed with these 9 terminations. Again, there are three voices- Active, Passive and Impersonal. Transitive verbs are used in the Active and Passive voices while intransitive verbs are conjugated in the Active and Impersonal voices. Addition of one or more of 22 prefixes (upasargas) to verb roots can result in more variety of forms. Derivative verb roots, both derived from verb roots as well as nominal words, also follow same process to form verb forms. There can be some specific rules and exceptions in some cases. A chart given here gives a rough estimate of all the possible verb-forms of Sanskrit. [48]



[48] **Mishra Sudhir K., Jha, Girish N.**, 2004, ***Identifying Verb Inflections in Sanskrit morphology***, In proc. of SIMPLE 04, IIT Kharagpur, pp. 79-81.

22(+) upasarga **22(+) upasarga**

The verb roots of *gaṇa* 1, 4, 6 and 10 adopt certain terminations when *tiṅ* affixes are added to them. Consequently, the verb roots of these class forms base ending in *'a'*. The *tiṅ* affixation also influences the verb root and it undergoes several morpho-phonemic changes such as its end vowel is gunated. The verb root can adopt certain more operations resulting in the final verb-form.

1.2.5 Survey of Morphological tools for Sanskrit

C-DAC

Deśika[49] (Natural Understanding System), a Software Package, developed by Indian Heritage Group, C-DAC, Bangalore led by P. Ramanujan, claims to generate and analyze plain and accented written Sanskrit texts using grammar rules of Pāṇini's Aṣṭādhyāyī, with an exhaustive database of Amarakośa and heuristics of semantics and contextual processing from Nyāya and Mīmāmsā Śāstra-s. It also claims to analyze Vedic texts. The analysis module of the Software is called as a general purpose Sanskrit parser which claims to dissolve and identify the compound and combined word forms, though the present version of the Software downloadable from the TDIL (Technology Development for Indian Languages) website has Subanta Generation Module only. This module has two modes: **choose mode** declines for the *prātipadika*-s already present in the list and **edit mode** declines for the entries which are not present in the list for which suitable gender and paradigm should be selected. Deśika is also supposed to include Vedic processing and śābdabodha[50].

---

[49] Deśika (Natural Language Understanding System), http://tdil.mit.gov.in/download/Desika.htm

[50] C-DAC R&D Activies: Development of Desika A Natural language Understanding (NLU) system for Sanskrit. [1990 - 1994], http://www.cdac.in/html/ihg/activity.asp, accessed on Jan 16, 2007

Sanskrit Authoring System (VYASA)[51] claims to be a robust multilingual document editor with transliteration facility among Indian Languages and Roman, sorting and searching facilities, indexing, and concordance. It also claims to provide tools for analyses at morphological, syntactic and semantic levels. Tools for searching/indexing/sorting, lexical updation, lexical tagging, extraction/indexing of quotations in commentaries/explanations, transliteration facility, word split programs for *sandhi* and *samāsa*, poetry analysis (textual/metric/statistical), statistical tools like concordance, thesauri, and electronic dictionaries are also included. However, the system is not available anywhere to evaluate or check.

Academy of Sanskrit Research, Melkote

Academy of Sanskrit Research (ASR), Melkote under the directorship of Prof. M. A. Lakshmitatachar has been working on NLP in Sanskrit and other Indian languages for more than 10 years. The software 'śābdabodha' developed at ASR, Melkote is available on TDIL website. It is claimed to be an interactive application built to analyze the semantic and syntactic structure of Sanskrit sentences. The software works on DOS 6.0 or higher with GIST (Graphic based Intelligence Script Technology) shell on Windows 95 platform. The software is said to include a user interface. For an input Sanskrit sentence or a file the program will check for the syntactic compatibility and otherwise gives all the morphological details with POS information having nouns, pronouns, adjectives, participles, indeclinables, indeclinable participle and verbs tags of the components of the sentence. It claims to process all types of sentences of Sanskrit, and can handle generation and analysis of *subanta* of more than 26,000 stems, *tiṅanta* conjugations of roots, in two voices, ten *lakāras* and three modes viz. *kevala tiṅanta, ṇijanta* and *sananta*. It also handles the generation and analysis and identification of case inflected forms of 11 types of *kṛdanta* of 150 roots. Apart from this it is said to have a database of 690 avyayas, 26,000 nominal stems, 600 verbal roots, *kṛdanata* forms of 600 verbal roots, 5 *taddhita* suffixes. For handling the semantic analysis, a matrix of 52 sets

[51] C-DAC, R&D Activities: Developing Sanskrit Authoring System (VYASA), http://www.cdac.in/html/ihg/activity.asp and The House Magazine of C-DAC, Pen to Paper Developing Sanskrit Authoring System – VYASA, http://www.cdac.in/html/connect/3q2000/art10a.htm

of nouns with their synonyms amounting to 300 nouns, 27 actions denoted by nearly 200 verbs are said to have been prepared.[52] This institution is also working with 20 odd software tools like **Bodha** (Sentence disambiguation system according to *śābdabodha* of *navīna nyāya* system), **Śemuṣī** (*subanta* generator/analyser), **Prajñā** (*tiṅanta* generator/analyser), **Cetanā** (*kṛdanta* generator/analyser), **Pāṇini** (*sandhi* joiner according to Pāṇinian rules) etc. which are to be released.[53]

Rashtriya Sanskrit Vidyapeetha (RSV), Tirupati

RSV Tirupati has put much effort in developing linguistic resources for NLP in Sanskrit. Prof. K.V. Ramakrishnamacharyulu, (presently V.C. of Rashtriya Sanskrit Sansthan, Jaipur) and Dr. Srinivasa Varkhedi along with Prof. Vineet Chaitanya and Amba P. Kulkarni have initiated many projects and have developed many tools like *pada-ccheda*, which isolates Sanskrit compound words into its components, which works on Sanskrit ISCII text in Linux environment. It is also working on developing Sanskrit morphological analyzer. An initial analyzer developed in collaboration with IIIT-H is already online. Apart from this it is also concentrating on *kṛdanta* and *tiṅanta* analyzers and also generators for *subanta, tiṅanta* and *samāsa*. RSV Tirupati along with C-DAC Bangalore, Ahobila Mutt Sanskrit College Madhurantakam Tamil Nadu, Poorna Prajna Samshodhana Mandiram Bangalore, Chinmaya International Foundation Veliyanad Kerala, ASR Melkote Karnataka, IIIT-H, Dept. of Sanskrit H.S.Gour University Sagar Madhya Pradesh have combined initiative to develop a large Sanskrit Corpus (presently this is not online).[54]

RCILTS – Utkal University

Resource Centre for Indian Language Technology Solutions (RCILTS) – Oriya Centre at the Department of Computer Science and Application, Utkal University, led by Prof.

[52] Language Processing Tools: TDIL website, http://tdil.mit.gov.in/nlptools/ach-nlptools.htm
[53] Academy of Sanskrit Research, Melkote, http://www.sanskritacademy.org/About.htm
[54] RSV Tirupati, http://rsvidyapeetha.ac.in and http://www.sansknet.org

Sangamitra Mohanty and funded by Ministry of Information Technology (MIT), has been working on Oriya-English-Hindi Trilingual Dictionary, Oriya and Sanskrit WordNet, Trilingual Word Processor, Grammar and Spell checkers and MorphAs for Oriya and also English to Oriya Machine Translation System. It has received IPR for four of their products from Ministry of Human Resource and Development (MHRD), Govt. of India.

San-Net (Sanskrit WordNet)

RCILTS-Utkal[55] claims to have developed a proto-type of Sanskrit WordNet using Navya-Nyāya and Pāṇinian Grammar. It has 300 Sanskrit words (250 Nominal words and 50 Verbal words) having synonymy, antonym, hyponymy, hypernymy, holonymy and meronymy relations with their analogy, etymology, and definitions. It also claims that a standard Knowledge Base (KB) that has been developed for analyzing syntactic, semantic and pragmatic aspects of any lexicon.

The Sanskrit Heritage Site

Dr. Gerard Huet, Director, INRIA has developed various computational tools for Sanskrit which are available online[56]. The **Declension Engine** declines all the nominal inflectional forms with the 'compound base' for a query word given with its gender information. It also gives the 'compound base' of the query word. The **Conjugation Engine** gives all the possible forms of the verb root in its *ātmane* and/or *parasmai* terminations, in *kartari* and *karmaṇi/bhāve* voices with its Desiderative forms in eight *lakāra*-s. It also gives few participle and indeclinable *kṛt* forms[57]. He claims that from 535 roots, his engine generates 14554 noun forms, 113935 root verbal forms, 203281 root participial forms, 14311 *iic* and periphrastic forms, and 737 indeclinable forms, totaling roughly half a million forms[58]. **Lemmatizer** and **Sanskrit Readers** are the analyzers. While the lemmatizer tries to tag a given simple inflected noun or a verb (without *upasarga*-s), the Sanskrit Reader Companion analyses a given phrase or a simple

[55] RC-ILTS – Computer Science & Application, Utkal University, http://www.ilts-utkal.org/sanskrit.htm
[56] The Sanskrit Heritage Site, Huet, Gerard. http://sanskrit.inria.fr/
[57] Ibid.

sentence, segments it into individual words, tags each word and parses the input. Modular transducers are applied to constraint the lexical analyzer to recognize the stream of forms as a regular expression over 14 phases, specifying Sanskrit's morphology geometry to get right composition of compound chunks. Over generation of segmentation and ambiguity of tags are checked by semantic role analysis similar to Pāṇini's kāraka theory and also by governance patterns of verbs.[59]

The Sanskrit Library

The Sanskrit Library Project, directed by Dr. Peter M. Scharf, Classics Dept., Brown University, has developed a web-based reading room holding Kramapāṭha, a Sanskrit independent-study reader on texts like Pañcatantra, Rāmopākhyāna and Pāṇini's Aṣṭādhyāyi. The reader gives the text in Devanagari with Roman transliteration, *sandhi* analysis, a detailed grammatical analysis and lexical analysis of the text, with notes and translation. As part of independent-study texts, a detailed classification of grammatical categories (tag set) is made.

A nominal inflection generator tool is also available which generates nominal inflections for a given word with gender and class information.[60]

IIIT Hyderabad

Language Technology and Research Centre (LTRC), IIIT-H, is one of the leading NLP centres in India. IIT Kanpur and later LTRC, IIIT-H, have been the pioneers in the field of language technology and to initiate Pāṇinian approach to NLP in India. LTRC has several ongoing activities with Govt. of India, Carnegie Mellon University's Language Technology Institute, UPENN, HP Labs, Google, Nokia and TCS besides several academic institutions in India. Prof. Rajeev Sangal and Prof. Vineet Chaitanya

---

[58] Huet, Gerard, 2006, Parsing Sanskrit by Computer, (abstract) in the Proceeding of the 13th World Sanskrit Conference, Edinburgh, UK

[59] Ibid.

[60] The Sanskrit Library Reading Room - http://cgi-user.brown.edu/cgi-user/sanskrit/login

with the Akshar Bharati[61] group have developed many NLP tools for Indian languages like MorphA-s (morphological analyzers), Anusaraka-s[62] (language access tool), Shakti Machine translation System. MorphA-s for Hindi, Telugu, Marathi, Kannada and Punjabi were developed as part of Anusaraka-s and now presently they are made available online and also can be downloaded from their website. Telugu MorphA is said to have coverage of 95% for any arbitrary modern standard Telugu text and Hindi MorphA has 88% coverage.[63] MorphA-s for these languages were developed based upon the paradigm model.[64] For a given word the MorphA gives the root word with its feature information like gender, number, person, tense etc.[65]

Sanskrit MorphA was developed by Vinish Jain, an M.Tech student of IIIT-H under the guidance of Amba P. Kularni, which was a further development of an earlier working Sanskrit MorphA developed at ASR Melkote. This MorphA was developed using paradigm approach in the model of other MorphA-s at IIIT-H which used a lexicon derived from Monier William's dictionary. This could only handle *subanta*-s. Later with the collaboration of RSVP Tirupati, the present Sanskrit MorphA[66] was developed by Amba P. Kulkarni and V.Sheeba. This MorphA has separate modules to handle *subanta*, *tiṅanta, kṛdanta* and *samāsa*. Each word is filtered through all the four modules and all possible answers are produced. *subanta* module contains 222 paradigms for nouns and pronouns with a root dictionary of around 1.5 M words extracted from Monier Williams dictionary. *tiṅanta* module has a database of ~10 M verb forms (no prefixed verbs). *kṛdanta* module can handle ~42 K of 20 types of *kṛdanta*. This approach has the advantage of developing further without disturbing the main program, as the program is

---

[61] Akshar Bharati, is a personification of groups (@ IITK, IIIT-H, University of Hyderabad, etc.) working on NLP with special emphasis to Indian Languages giving due attention to Indian theories of grammar and language.

[62] Anusaaraka or a Language Accessor is a computer software which renders text from one Indian language into another (Kannada-Hindi, Marathi-Hindi, Punjabi-Hindi and Telugu-Hindi). It produces output not in the target language but close to it, which is in comprehensible to the reader who is trained to read the output.

[63] Morphological Analysers – IIIT Hderabad – http://www.iiit.net/ltrc/morph/morph_analyser.html

[64] Bharati, Akshar et al, 1999, Natural Language Processing: A Paninian Perspective, Prentice Hall Pvt. Ltd. New Delhi, pp.39-43.

[65] Ibid..

[66] RSVP, http://rsvidyapeetha.ac.in/~anusaraka (accessed: 13.10.2006)

independent of data and also more modules can be added.[67] This MorphA can handle only *sandhi*-free text and cannot filter the multiple tags and disambiguate it for a single word in a given context.

Jawaharlal Nehru University (JNU)

The RCILTS – Sanskrit, Japanese, Chinese unit of JNU, under the leadership of Prof. G.V.Singh claims to have developed web based Sanskrit Language Learning System for the use of scholars for designing Knowledge based systems based on the Indian traditions. The unit has developed a computational module of Aṣṭādhyāyī of Pāṇini, Sanskrit-English lexicon, English-Sanskrit lexicon and a lexicon of Nyāya terms. It also says that it has made some efforts on the *sandhi* analysis system.[68]

Girish Nath Jha[69] developed a Nominal Inflection Generator for Sanskrit for his M. Phil. dissertation. The program, written in prolog, generates all the inflections of *subanta* given a Sanskrit word with gender and ending letter information.

Special Centre for Sanskrit Studies, JNU:

Special Centre for Sanskrit Studies, JNU, established in 2002, has undertaken the task of developing NLP tools for Sanskrit. Under the guidance of Dr. Girish Nath Jha, the students have developed certain useful programs for computational processing of Sanskrit texts. A list of works is given below:

The NLP team at the centre has developed an Online Multilingual Amarakosha (OMA) under a project funded by UGC under UPOE program. It is Unicode based software which supports seven languages- Sanskrit, Hindi, Kannada, Punjabi, Bangla, Oriya and English and allows the user to search the synonym from one language to another. The

---

[67] Bharati Akshar, Amba Kulkarni, V Sheeba, "Building a Wide Coverage Sanskrit Morphological Analyzer: A Practical Approach", in the Proceedings of First National Symposium on Modelling and Shallow Parsing of Indian Languages, 2-4 April 2006, IIT Bombay, Mumbai

[68] RCILTS, JNU – Achievements: http://tdil.mit.gov.in/SanskritJapaneseChinese-JNUJuly03.pdf (accessed 15.10.2006)

[69] Jha, Girish Nath, 1993, 'Morphology of Sanskrit Case Affixes: A Computational Analysis', M.Phil., submitted to JNU, New Delhi.

output displays the grammatical and semantic category of the word, its base word, reference and ontological information. The software also provides the facility to enter and edit the data by language experts. The software is expected to be extended as a multilingual interface, search engine and text processing tool in near future.[70]

R. Chandrashekhara[71], developed tag-sets for POS tagging of Sanskrit text as part of his Ph.D. thesis. This tagger (POST) which is an online system run on Apache Tomcat platform using Java Servlet takes *sandhi*-free classical Sanskrit prose text as input and provides the tagged text as output. The system is very important for the further R&D on the Sanskrit-Indian Languages Machine Translation Systems (MTS).

Sudhir Kumar Mishra[72], has recently completed his Ph.D. research on Kāraka Analyzer for Laukika Sanskrit prose text based on Pāṇini's Kāraka formulations. Kāraka rules are of central importance in Sanskrit syntactic structure. As part of this research work he also worked on identification of verb inflections in Sanskrit morphology[73].

Subash Chandra[74] developed a Sanskrit *Subanta* Recognizer and Analyser System (SRAS) for his M.Phil. dissertation. SRAS is also an online system run on Apache Tomcat platform using Java Servlet and MSSQL server 2005 as back end. This system has been developed according to Pāṇinian formulation which accepts only non-joint (*sandhi-rahita*) Sanskrit text in Devanāgarī script and fully depends on both the rule base, example base and a database of other linguistic resources. The system has been tested with some selected corpora and claims to give an average accuracy of 91.65%.

Research work is also being done on learning Sanskrit language using e-learning approach[75], and *sandhi* analyzer applying Pāṇinian and some heuristic rules[76], online

---

[70] System is accessible @ http://sanskrit.jnu.ac.in

[71] Chandrashekhar, R, 2006, 'POS Tagging for Sanskrit', submitted for Ph.D degree at SCSS, JNU.

[72] Mishra, Sudhir Kumar, 2007, 'Sanskrit Karaka Analyzer for Machine Translation', submitted for Ph.D. degree at SCSS, JNU

[73] Mishra, Sudhir Kumar & Girish Nath Jha, 2005, 'Identifying Verb Inflections in Sanskrit Morphology', in the proceedings of SIMPLE-05, IIT-Kharagpur, pp.79-81

[74]Chandra, Subash, 2006, 'Machine Recognition and Morphological Analysis of Subanta-padas', submitted for M.Phil degree at SCSS, JNU

[75] Bhowmik, Preeti & Jha, Girish Nath, 2006, 'Sanskrit Language Pedagogy: an e-learning approach', In the Souvenir Abstracts of 28th AICL, BHU, Varanasi, p.150.

indexing of Ādiparva of Mahābhārata[77], computational analysis of Sanskrit gender and analysis of derived nouns in Sanskrit.[78]

Sanskrit to English Translator:

Aparna Subramanian of School of Computer Science, Devi Ahilya Vishwavidyalaya, Indore worked on a project titled "Sanskrit to English Translator"[79] for her M.Sc. degree. It performs sandhi viccheda of Sanskrit words and then translates them to English. It has two components: morphological parser and translation generator. The morphological parser module consists of a set of transducers that transform the provided input text to a set of acceptable output. It also gives the parse of these words. The second part of the parser is a Viccheda module, which applies reverse sandhi rules to split the combined Sanskrit words into separate basic words. The translator is also a two part module, which first structures the English sentence according to the grammar using the parse information. The second generates equivalent English words according to the morphological details. The two combined together give the needed translated sentence.

---

[76] Kumar, Sachin & Jha, Girish Nath, 2006, 'Issues in sandhi processing of Sanskrit', In the Souvenir Abstracts of 28th AICL, BHU, Varanasi, p.129.

[77] Mani, Diwakar, & Jha, Girish Nath, 2006, 'Online indexing of Ādiparva of Mahābhārata', In the Souvenir Abstracts of 28th AICL, BHU, Varanasi, p. 125.

[78] Singh, Surjit Kumar & Jha, Girish Nath, 2006, 'Strategies for Identifying and Processing Derived Nouns in Sanskrit', In the Souvenir Abstracts of 28th AICL, BHU, Varanasi, p. 131.

[79] Subramanian, Aparna, Jan 2005, "Sanskrit to English Translator", Language in India, Vol. 5:1.

Chapter 2

Sanskrit Verb Morphology

2.1.0 Sanskrit verb-roots: dhātupāṭha (DP)

Sanskrit verb forms are derived, as traditional analysis follows, from verb roots. These verb-roots are called *dhātus*-the meaning-bearing entity. This concept of *dhātu* has a practical use in explaining the language through its analysis without implying that the dhātus afford the ultimate explanation.The discovery of *dhātus* is supposed to take place long before Pāṇini compiled his Dhātupāṭha-the catalogue of dhātus. There are several dhātupāṭhas but the one which is compiled by Pāṇini is considered to be the oldest among available dhātupāṭhas. The pattern followed in the Pāṇinian dhātupāṭha is mostly followed by the latter grammarians.

In Pāṇinian DP., roots are listed alongwith their meanings. These two – root and its meaning- form a sentence, usually known as *dhātusūtra*. Generally only one meaning is assigned to a root and is given in the form of an action noun in the locative case. For example: *bhū sattāyām, edha vṛddhau* etc. Sometimes, however, it is given in the form of a *bahuvrīhi* compound with – artha as the second member, such as *gatyartha, śabdārtha* etc. The root is, further, accompanied by one or more *anubandhas* or the code letters indicating certain features of the root.

With reference to certain peculiar modifications (*vikaraṇas*) which the roots undergo before the terminations of the two tenses -present and imperfect- and the two moods - imperative and potential- Sanskrit verbs are divided into ten classes (*gaṇas*) called conjugations. Each class or conjugation has been named after the root it begins with. The ten classes with related information are listed in the table produced here:

S. No.	Class (gaṇa)	First root	Vikaraṇa	Characteristic	Number of roots
1	***bhvādigaṇa***	*bhū*	*śap*	*a*	1035
2	***adādigaṇa***	*ad*	*luk~śap*	-	71
3	***juhotyādi***	*hu*	*ślu~śap*	*u*	24
4	***divādi***	*div*	*śyan~śap*	*a*	141
5	***svādi***	*su*	*śnu~śap*	*nu*	34
6	***tudādi***	*tud*	*śa*	*a*	155
7	***rudhādi***	*rudh*	*śnam*	*na*	25
8	***tanādi***	*tan*	*u*	*u*	10
9	***kryādi***	*krī*	*śnā*	*nā*	62
10	***curādi***	*cur*	*ṇic*	*ya*	410

The characteristics (*vikaraṇas*) of the ten classes or conjugations of verbs appear only in the two tenses and the two moods named above. These four are termed as special or conjugational tenses and moods whereas remaining six are called general or non-conjugational.

The list is further arranged into groups that share common accent-patterns, which can be of following types:

a. the stem may be accented
b. the stem may be unaccented in the list and again
    (i) the group classifier vowel may be accented, or
    (ii) the group classifier vowel may be unaccented, or
    (iii) the group classifier vowel may have circumflex accent.

Sometimes monosyllabic stems ending in vowels are classified by consonant classifiers if they have a distinctive bunch of functions in terms of morphemic distribution. The arrangement within such a sub-group of lists sharing common accent-patterns is not in any alphabetical order, but quite often stems sharing the same vowel classifier are grouped together, as also phonemically similar items like kūrd, khūrd A, and gūrd A. There is specific functions of each classifier.

Verb roots or stems can also be sub-classified by following classifiers:

I. non-*i*-augmenting (*aniṭ*), which include

- **a.** unaccented monosyllabic stems (listed in DP)[1]
- **b.** other stems under specific environments[2]

(i) Exclusively middle ending (*ātmanepada*)

- **a.** Unaccented vowel- classified and -Ṅ classified[3]
- **b.** All stems under passive[4]
- **c.** Other stems in specified environments enumerated in[5]

(ii) Middle ending (only if the action is for the subject) otherwise active ending (*parasmaipada*) class

- **a.** Circumflex accented vowel classified and *ñ*-classified stems[6]
- **b.** Other stems in specified environments enumerated in[7]

(iii) Exclusively active ending class

- **a.** By elimination the rest, i.e., the acute accented vowel classified and others[8]
- **b.** Other stems in specified environments enumerated in[9]

II. *i*-augmenting (*seṭ*)

- **a.** the residue of *i* under non -*i*- augmenting 7.2.10
- **b.** other stems in specified environments enumerated in[10]

These stems can be also classified among (i), (ii) and (iii) as above.

III. Optionally –*i*- augmenting (*veṭ*)

- **a.** Stems enumerated in[11]

These stems can be also classified among (i), (ii) and (iii) as above.

---

[1] *ekāca upadeśe anudāttāt* 7/2/10
[2] *Aṣṭādhyāyī*: 7/2/11-26, 7/2/59-65
[3] *anudātta ṅita ātmanepadam* 1/3/12
[4] *bhāva-karmaṇoḥ* 1/3/13
[5] *Aṣṭādhyāyī*: 1/3/14-71
[6] *svarita-ñitaḥ kartrabhiprāye kriyā-phale* 1/3/72
[7] *Aṣṭādhyāyī*: 1/3/73-77
[8] *śeṣāt kartari parasmaipadam* 1/3/78
[9] *Aṣṭādhyāyī*: 1/3/79-93
[10] *Aṣṭādhyāyī*: 7/2/35-36, 7/2/52-54, 7/2/58, 7/2/66-78
[11] *Aṣṭādhyāyī*: 7/2/27-34, 7/2/37-51, 7/2/55-57

2.1.1 Lakāras-Tenses and Moods:

Sanskrit has six tenses (*kāla*) and four moods (*artha*) technically termed as ten ***lakāras***.

Tenses are as follows:

laṭ[12] or *bhavantī vṛttiḥ* (Present - *vartamānaḥ*)

Three kinds of past tenses-

laṅ[13] or *hyastanī vṛttiḥ* (Imperfect - *anadyatanabhūtaḥ*)

liṭ[14] or *parokśavṛttiḥ* (Perfect - *parokṣabhūtaḥ*)

luṅ[15] or *adyatanī vṛttiḥ* (Aorist - *bhūtaḥ*)

Two kinds of future tenses-

luṭ[16] or *śvastanī vṛttiḥ* (1st Future - *anadyatanabhaviṣyat*)

lṛṭ[17] or *bhaviṣyantī vṛttiḥ* (2nd Future - *bhaviṣyat*)

Moods:

loṭ[18] (Imperative - *ājñā*)

vidhiliṅ[19] or *saptamī vṛttiḥ* (Potential - *vidhiḥ*)

āśīrliṅ[20] or *āśīrvṛttiḥ* (Benedictive - *āśīḥ*)

lṛṅ[21] or *kriyātipattiḥ* (Conditional - *saṅketaḥ*)

Seven of these *lakāra*-s are of common occurrence ; viz. 1. the present, 2. the imperfect (also called the first preterite), 3. the potential (or optative), 4. the imperative, 5. the perfect (also called the second preterite), 6. the first future, 7. the second future. Three are not so commonly used; viz. 8. the aorist (also called the third preterite), 9. the precative (also called the benedictive), 10. the conditional.[22]

---

[12] *varttmāne laṭ* 3/2/123
[13] *anadyatane laṅ* 3/2/111
[14] *parokṣe liṭ* 3/2/115
[15] luṅ 3/2/110
[16] *anadyatane luṭ* 3/3/15
[17] lṛṭ śeṣe ca 3/3/13
[18] *loṭ ca* 3/3/162; *āśiṣi liṅloṭau* 3/3/173
[19] *vidhinimantraṇāmantraṇādhīṣṭasampraśnaprārtheṣu liṅ* 3/3/161
[20] *āśiṣi liṅloṭau* 3/3/173
[21] *liṅnimitte lṛṅ kriyātipattau* (3/1/139); *bhūte ca* (3/3/140)
[22] Whitney

There is also an infinitive mood, and several participles. Of these, the present, the three past tenses, and the two futures belong to the indicative mood.

The **imperfect** refers to an event done at some time in recent past, but before the current day. It may denote action past or continuing. The **perfect** refers to an event completely done before the present day at some remote period, unperceived by or out of sight of the narrator. The **aorist** refers to an event done and past at some indefinite period, whether before or during the current day. The *luṭ* or **first future** expresses definite futurity. *lṛṅ* or **second future** expresses indefinite futurity. The **potential** expresses the sense of commanding, directing, inviting, expression of wish, enquiring, requesting. The **conditional** is occasionally used after the conjunctions *yadi* and *cet*, 'if'. The **precative** or **benedictive** is used in praying and blessing.

2.1.2 Terminations:

The terminations affixed to the verb roots-primitive or derived- to produce verb forms are called *tiṅ* terminations. These are a set of 18 terminations, termed as *tiṅ* following the pratyāhāra technique of Pāṇini.[23] The terminations are divided into two parts, each consisting of 9 in two different padas: Parasmaipada and Ātmanepada. In each group nine terminations are result of multiplication of three persons to three numbers.[24]

	Parasmaipada			**Ātmanepada**		
	Singular	**Dual**	**Plural**	**Singular**	**Dual**	**Plural**
3rd person -	*tip*[25]	*tas*	*jhi*	*ta*[26]	*ātām*	*jha*
2nd person -	*sip*	*thas*	*tha*	*thas*	*āthām*	*ḍhvam*
1st person -	*mip*	*vas*	*mas*	*iṭ*	*vahiṅ*	*mahiṅ*

The terminations listed here have certain memorial letters such as *p* in *tip*, *sip* etc. These letters serve special purposes in the declension process. Each of the set of 9 is replaced

---

[23] *tip-tas-jhi-sip-thas-tha-mip-vas-mas-ta-ātām-jha-thās-āthām-dhvam-iḍ-vahiṅ-mahiṅ* 3/4/78

[24] *tiṅastrīṇi trīṇi prathamamadhyamottamāḥ* 1/4/100;
tānyekavacana-dvivacana-bahuvacanānyekaśaḥ 1/4/101

[25] *laḥ parasmaipadam* 1/4/98

[26] *taṅānāvātmanepadam* 1/4/99

with corresponding forms in different lakāras. The terminations thus obtained are again modified so that they can be affixable to the verb roots.

The parasmaipada is that inflected word or verb (*pada*) the action of which is addressed to another than the agent (*parasmai*, dative of *parā* 'another'). Ātmanepada is a word or verb the action of which is addressed or reverts to the agent himself (*ātmane*, dative of *ātman*, 'self'). These might be rendered, therefore, 'transitive' and 'reflective' verbs, but in a peculiar sense that the action is said to affect either a different agent or the agent himself; it is the result rather than the action, and this is therefore compatible with an intransitive verb. Thus, in *'devadattaḥ odanaṁ pacati'* where the root *pac* (to cook) is conjugated in *parasmaipada,* the sense denoted is 'Devadatta cooks for his master/ or someone else'; but in *'devadattaḥ odanaṁ pacate'* where the root is in *ātmanepada* it means 'Devadatta cooks for himself'. In ordinary usage, however, the distinction import is little observed. Some verbs are conjugated in one voice, some in the other, and some in both, without much attention being paid to their signification or relations. Pāṇini formulates some rules for the determination of *ātmanepada.*

The four tables given here lists all the *tiṅ* terminations of *parasmaipada* and *ātmanepada* in ten *lakāra*-s. The column of terminations lists the basic set of tiṅ terminations. The column-I lists, separately for each lakāra, the terminations which are replacements of basic *tiṅ* suffixes. The terminations listed in column-II are modified terminations which are appended as such to the verb roots.

Table-1(a) The list of terminations for parasmaipada in first five lakāras

Termi.	Present (laṭ)		Perfect (laṅ)		Potential (vidhiliṅ)		Imperative (loṭ)		Imperfect (liṭ)	
	I	II	I	II	I	II	I	II	I	II
तिप्	तिप्	ति	दिप्	त्/द्[27]	यात्[28]	इत्[29] यात् 2,3,7,5,8,9	तुप्[30]	तु	णल्[31]	अ
तस्	तस्	तः	ताम्[32]	ताम्	याताम्	याताम् इताम्	ताम्[33]	ताम्	अतुस्	अतुः
झि	अन्ति[34]	न्ति 1,4,6,10 अन्ति। अति	अन्	न्, 1,4,6,10 अन्, ऊः	युस्[35]	इयुः 1,4,6,10 युः/	अन्तु	न्तु 1,4,6,10 अन्तु/ अतु	उस्	ऊः
सिप्	सिप्	सि	सिप्	स्	यास्	इः/ याः/	हि[36]	-[37]/हि/ धि/ -आन	थल्	इथ[38] / थ
थस्	थस्	थः	तम्	तम्	यातम्	इतम्/ यातम्	तम्	तम्	अथुस्	अथुः
थ	थ	थ	त	त	यात	इत/ यात/	त	त	अ	अ
मिप्	मिप्	मि	अमप्	म्/अम्	याम्	इयम्/ याम्/	आनिप्[39]	आनि	णल्	अ
वस्	वस्	वः	व[40]	व	याव	इव/ याव/	आवप्	आव	व	इव
मस्	मस्	मः	म	म	याम	इम/ याम/	आमप्	आम	म	इम

[27] *itaśca* 3/4/99
[28] *yāsuṭ parasmaipadeṣūdātto ṅicca* 3/4/103; *liṅaḥ salopo'nantyasya* 7/2/79
[29] *ato yeyaḥ* 7/2/80; *lopo vyorvali* 6/1/64
[30] *eruḥ* 3/4/83
[31] *parasmaipadānāṁ ṇalatususthalathusaṇalvamāḥ* 3/4/82
[32] *tas-thas-tha-mipāṁ tān-tan-tāmaḥ* 3/4/101
[33] *loṭo laṅvat* 3/4/85
[34] *jho'ntaḥ* 7/1/3
[35] *jherjus* 3/4/108
[36] *serhyapicca* 3/4/87
[37] *ato heḥ* 6/4/105
[38] *ārdhadhātukasyeḍvalādeḥ* 7/3/35
[39] *merniḥ* 3/4/89; *āḍuttamasya picca* 3/4/92
[40] *nityaṁ ṅitaḥ* 3/4/99

Table-1(b) The list of terminations for parasmaipada in other five lakāras

Term i.	First Future (luṭ)		Second Future (lṛṭ)		Aorist (luṅ)		Benedictive (āśīrliṅ)		Conditional(lṛṅ)	
	I	II	I	II	I	II	I	II	I	II
तिप्	ता[41 42]	ता	स्यति[43]	स्यति	सीत्	सीत्/ईत्/ अत्, त्	यात्[44]	यात्	स्यत्	स्यत्
तस्	तारौ[45]	तारौ	स्यतस्	स्यतः	स्ताम्	स्ताम्।ताम् इष्टाम्/ अताम्। ताम्	यास्ताम्	यास्ताम्	स्यताम्	स्यताम्
झि	तारस्	तारः	स्यन्ति	स्यन्ति	सुः	सुस्/ इषुस्/ अन्,उस्	यासुस्	यासुः	स्यन्	स्यन्
सिप्	तासि[46]	तासि	स्यसि	स्यसि	सीः	सीस्/ ईस्/ अस्,स्	यास्	याः	स्यस्	स्यस्
थस्	तास्थस्	तास्थः	स्यथस्	स्यथः	स्तम्	स्तम्, तम्/ इष्टम्/ अतम्।तम्	यास्तम्	यास्तम्	स्यतम्	स्यतम्
थ	तास्थ	तास्थ	स्यथ	स्यथ	स्त	स्त,त/ इष्ट/ अत,त	यास्त	यास्त	स्यत	स्यत
मिप्	तास्मि	तास्मि	स्यामि	स्यामि	सम्	सम्/ इषम्/ अम्	यासम्	यासम्	स्यम्	स्यम्
वस्	तास्वस्	तास्वः	स्यावस्	स्यावः	स्व	स्व/इष्व/ आव/व	यास्व	यास्व	स्याव	स्याव

[41] *sya-tāsī lṛ-luṭoḥ* 3/1/33
[42] *luṭaḥ prathamasya ḍā-rau-rasaḥ* 2/4/85
[43] Ibid.
[44] *liṅaḥ salopo'nantyasya* 7/2/79
[45] *ri ca* 7/4/51
[46] *tāsastyorlopaḥ* 7/4/50

मस्	तास्मस्	तास्मः	स्यामस्	स्यामः	स्म	स्म/इष्म/ आम,म	यास्म	यास्म	स्याम	स्याम

Table-2(a) The list of terminations for ātmanepada in first five lakāras

Termi.	Present (laṭ)		Perfect (laṅ)		Potential (vidhiliṅ)		Imperative (loṭ)		Imperfect (liṭ)	
	I	II	I	II	I	II	I	II	I	II
त	ते[47]	ते	तन्	त	ईत[48]	ईत	ताम्[49]	ताम्	ए[50]	ए
आताम्	आते	इते आते	आताम्	इताम् आताम्	ईयाताम्	ईयाताम्	आताम्	इताम् आताम्	आते	आते
झ	अन्ते	अते अन्ते	अन्त	न्त अत	ईरन्[51]	ईरन्	अन्ताम्	अताम् न्ताम्	इरे	इरे
थास्	से[52]	से	थास्	थाः	ईथास्	ईथाः	स्व[53]	स्व	से	इषे
आथाम्	आथे	इथे/ आथे	आथाम्	इथाम् आथाम्	ईयाथाम्	ईयाथाम्	आथाम्	इथाम् आथाम्	आथे	आथे
ध्वम्	ध्वे	ध्वे	ध्वम्	ध्वम्	ईध्वम्	ईध्वम्	ध्वम्	ध्वम्	ध्वे (ढ्वे[54])	इध्वे(इढ्वे)
इट्	ए	इ/ ए	इ	इ	ईय[55]	ईय	ऐप्[56]	ऐ	ए	ए
वहिङ्	वहे	वहे	वहि	वहि	ईवहि	ईवहि	आवहैप्	आवहै	वहे	इवहे

[47] *ṭita ātmanepadānāṁ ṭere* 3/4/79
[48] *liṅaḥ sīyuṭ* 3/4/102
[49] *āmetaḥ* 3/4/90
[50] *liṭastajhayorośireca* 3/4/81
[51] *jhasya ran* 3/4/105
[52] *thāsaḥ se* 3/4/80
[53] *savābhyāṁ vā'mau* 3/4/91
[54] *iṇaḥ ṣīdhvaṁluṅliṭāṁ dho'ṅgāt* 8/3/78
[55] *iṭo't* 3/4/106
[56] *eta ai* 3/4/93

महिङ्	महे	महे	महि	महि	ईमहि	ईमहि	आमहैप्	आमहै	महे	इमहे

Table-2(b) The list of terminations for Ātmanepada in other five lakāras

Termina -tions	First Future (luṭ)		Second Future (lṛṭ)		Aorist (luṅ)		Benedictive (āśīrliṅ)		Conditional (lṛṅ)	
	I	II	I	II	I	II	I	II	I	II
त	ता	ता	स्यते	स्यते	स्त	स्त	सीष्ट[57]	सीष्ट इष्ट अत	स्यत	स्यत
आताम्	तारौ	तारौ	स्येते	स्येते	साताम्	साताम्	सीया-स्ताम्	सीयास्ताम् इषाताम् एताम्/ आताम्	स्येताम्	स्येताम्
झ	तारस्	तारः	स्यन्ते	स्यन्ते	सत	सत	सीरन्	सीरन् इषत[58] अन्त/अत	स्यन्त	स्यन्त
थास्	तासे	तासे	स्यसे	स्यसे	स्थास्	स्थाः/ थाः	सीष्ठास्	सीष्ठाः इष्ठाः अथाः	स्यथास्	स्यथाः
आथाम्	तासाथे	तासाथे	स्येथे	स्येथे	साथाम्	साथाम्	सीया-स्थाम्	सीयास्थाम् इषाथाम् एथाम्/ आथाम्	स्येथाम्	स्येथाम्
ध्वम्	ताध्वे[59]	ताध्वे	स्यध्वे	स्यध्वे	ध्वम् (ढ्वं[60])	ध्वम् (ढ्वं)	सीध्वम्	सीध्वम् इध्वम् अध्वम्	स्यध्वम्	स्यध्वम्
इट्	ताहे[61]	ताहे	स्ये	स्ये	सि	सि	सीय	सीय इषि ए/ इ	स्ये	स्ये

[57] *suṭ tithoḥ* 3/4/107
[58] *ātmanepadeṣvanataḥ* 7/1/5
[59] *dhi ca* 8/2/25
[60] *iṇaḥ ṣīdhvaṁluṅliṭāṁ dho 'ṅgāt* 8/3/78
[61] *ha eti* 7/4/52

वहिङ्	तास्वहे	तास्वहे	स्यावहे	स्यावहे	स्वहि	स्वहि	सीवहि	सीवहि इष्वहि आवहि	स्यावहि	स्यावहि
महिङ्	तास्महे	तास्महे	स्यामहे	स्यामहे	स्महि	स्महि	सीमहि	सीमहि इष्महि आमहि	स्यामहि	स्यामहि

These terminations are supposed to be applicable to all verbs, whether primitive or derivative: and as in noun, so in verbs, the theory of Indian grammarians is, that before these terminations can be affixed to roots, an inflective base must be formed out of the root. Ten different rules, therefore, are propounded for forming verbal bases out of roots in the first four tenses; while all verbs are arranged under ten classes, according to the form of the base required by one or other of these rules. In the other tenses there is one general rule for forming the base, applicable to all verbs of whatever class.

2.1.3 Passive forms:

In Sanskrit, the passive form varies entirely in the conjugational tenses from that of the active verb (unless that verb belong to the 4th conjugation), while the terminations may sometimes be the same, as those of *ātmanepada.* It is rather a distinct derivative from a root, formed on one invariable principle, without any necessary community with the conjugational structure of the active verb. Thus, the root *dviṣ* (to hate) of the 2nd class makes *dveṣṭi* or *dviṣṭe* (he hates); another root *bhid* (to divide) of the 7th class makes *bhinatti* or *bhinte* (he divides); but the passive forms of both the roots is formed according to one invariable rule, by the simple insertion of *ya*, without reference to the conjugational form of the active, i.e. *dviṣyate* (he is hated) and *bhidyate* (he is divided) respectively.

In fact, though it is a distinct form of the root, a passive verb is really nothing but a verb conjugated according to the rule for the 4th class restricted to the *ātmanepada.* To say that every root may take a passive form is to say that roots of the 1st, 2nd, 3rd, 5th, 6th, 7th, 8th, 9th and 10th classes may all belong to the 4th, when they a yield a passive sense: so that if a root be already of the 4th class, its passive form is frequently identical in form with its

own *ātmanepada*. The only difference is that the accent in the former is on the syllable *ya* while in the *ātmanepada* of the primitive root, it is on the radical syllable.

2.1.4 Causal Verbs

Any root in every one of the ten classes may take a causal form, which is conjugated as a verb of 10th class in all the tenses and moods of the three voices. Causal forms are created by adding the suffix *ṇic* to the roots, thus being called *ṇijanta* (ending in *ṇic* suffix). Causal forms are primarily employed to give a causal sense to a primitive verb, i.e. the sense that a person or thing causes another person or thing to perform the action or to undergo the state denoted by the root. In some cases, causal form also supplies an active sense to a neuter verb. For example, *paṭhati* '(he) reads' becomes *pāṭhayati* '(he) causes to read' or '(he) teaches', and the neuter verb *kṣubhayati* 'he shakes,' 'is shaken' (from *kṣubh*) becomes *kṣobhayati* 'he shakes' (actively). Sometimes, causal forms denote other analogous senses. For example, *hārayati,* 'he allows to take', *nāśayati,* 'he suffers to perish;' *kṣamayati,* 'he asks to be forgiven'.

The terminations added to the verbal base of the causal forms are same as those of the primitive verbs. The root, however, undergoes certain changes before terminations are added to it. The suffix *ṇic* converts to *aya* that is added to every root in order to create a causal form.

2.1.5 Desideratives

A root in any of the ten classes may take a desiderative form. Some primitive roots also take desiderative form, though they not give the sense of desiderative. For ex. *jugups* (to blame) from *gup*, *cikits* 'to cure' from *kit*. A desiderative form yields the sense that a person or thing wishes to perform the action or is about to undergo the state indicated by

the desiderative form. For ex. *bodhati* (from *budh*, to know) he knows, *'bubodhiṣati'* he wishes to know.

To derive a desiderative base, the root is first reduplicated according to the general rules of reduplication. Then, termination *s* is added on to it in the non-conjugational tenses and moods. In conjugational *lakāras*, *a* is added to *s*.
The desiderative verb-base takes the same terminations of *parasmaipada* or *ātmanepada* or both as the primitive root does.

2.1.6 Frequentatives

Roots that contain single vowel and start with a consonant,[62] can adopt frequentative forms. The frequentative implies repetition or heightens the idea contained in the root. It, however, is of two kinds: the first one is a reduplicated Ātmanepada verb, with ya-affixed, confirming, like neuter and passive verbs, to the conjugation of the 4th class, and usually yielding a neuter signification. The other one is a reduplicated parasmaipada verb, following the conjugation of third class of the verbs. The terminations for the former will be those of Ātmanepada and for the latter, they will be the regular parasmaipada terminations.

2.1.7 Rules for formation of verbal bases in general:

1st Class (bhvādigaṇa)

The vowel of the root is replaced with its *guṇa* vowel (unless it be *a*, or a long vowel not final, or a short vowel followed by a double consonant) before every termination of the four tenses, and affix *a* is added to the root thus gunated. This vowel *a* is lengthened into *ā* before the initial *m* and *v* of a termination, but not when *m* is final, as in the 1st sing imperfect parasmaipada.

2nd Class (adādigaṇa)

[62] *dhātorekāco halādeḥ kriyāsamabhihāre yaṅ* 3/1/22

The vowel of the root is gunated (if it is, as above, capable of *guṇa*) before those terminations only which are marked with *parasmai.* Before all other terminations, the original vowels of the root are retained.

3rd Class (juhotyādigaṇa)

The initial consonant and vowel is reduplicated, and the radical is gunated and not the reduplicated vowel before the *parasmai* terminations only, as in the 2nd conjugation.

4th Class (divādigaṇa)

ya—lengthened to *yā* is affixed before initial *m*[63] and *v*-- to the root, the vowel of which is generally left unchanged.

5th Class (svādigaṇa)

nu is affixed to the root, and *nu* is gunated to *no* before the parasmaipada terminations only.

6th Class (tudādigaṇa)

a—which is lengthened to ā before initial *m** and *v*—is affixed to the root. It, however, does not change in other environments.

7th Class (rudhādigaṇa)

na is inserted between the vowel and final consonant of the root before the parasmaipada terminations, and *n* before the other terminations. The peculiarity of this conjugation is that the conjugational *na* or *n* is inserted into the middle of the root, and not affixed.

8th Class (tanādigaṇa)

u is added to the root, and this *u* is guṇated to *o* before the parasmaipada terminations only. As all the roots, except one, in this class, end in *n* , the 8th conjugation appears similar to the 5th.

9th Class (kryādigaṇa)

[63] but not before initial m of the termination of the 1st sing. Imperfect perfect.

nā is affixed to the root before the parasmaipada terminations; *nī* before all the others, except those beginning with vowels, where only *n* is affixed.

10th Class (curādigaṇa)

The radical vowel (if capable of guṇa) is substituted with its guṇa throughout all the persons of all the tenses. *aya*—lengthened to *ayā* is affixed before initial *m* and *v*—to the root thus guṇated.

An examination of the above listed rules shows that the object of all of them, except the 2nd, 3rd and 7th classes, is to insert a vowel, either alone or preceded by y or n, between the modified root and the terminations; and that the 1st, 4th, 6th and 10th, agree in requiring that the vowel, which is immediately to precede the terminations, shall be *a* or *ā*. It will appear, moreover, that the 2nd, 3rd and 7th, alone agree in not interposing a vowel between the final of the root and the terminations; and that the 5th, 8th and 9th classes agree in interposing either *u*, *ā* or *ī*, after the letter *n*. However, the conjugational characteristic has reference only to the four conjugational tenses (except only in the tenth conjugation), and that in the other tenses the base is formed according to one general rule for all verbs of whatever class; or, in other words, that in these tenses all verbs, of whatever class, are as if they belonged to one common conjugation.

2.2 The roots of first class-*bhvādigaṇa*:

The 1st class of verb roots which is also called *bhvādigaṇa*, includes more than half of the total roots of *dhātupāṭha*. Roots of this class commonly adopt *a* as characteristic. This *a* is remaining of characteristic *śap*. The initial and ending vowels are deleted. This characteristic, however, is only applied in the conjugational tenses and moods before the terminations are added, and not in all tenses and moods.

The ending vowel of roots short or long and the penultimate vowel substitute their guṇa before the characteristic *a* of this conjugation. Thus *i/ī* becomes *e*; *u/ū* replaced by *o*; *ṛ/ṝ* change to *ar*; *ḷ* to *al*.

2.2.1 Rules to form verb forms (of *bhvādigaṇa*) in ten *lakāras*:

This section describes general rules to form verbal bases from verb roots in Sanskrit in all the tenses. The description, though takes into account verb roots all the 10 classes (gaṇas), focuses on the roots of 1st classes for the sake of this dissertation.

The present tense (laṭ):

The final *a* of the base (*aṅga*) is lengthened before any conjugational termination beginning with a semi-vowel, nasal, or the letter *h, jh, bh.*

When the final *a* of a base is followed by *a, e* or *o*, the latter vowel is substituted for both.

When *i-u-ṛ-ḷ* (short/long) follow *a/ā*, the substitute for both is corresponding *guṇa* letter.

Examples: *bhū→bho; hṛ→har; budh→bodh; nī→nay*

Some roots of *bhvādigaṇa* undergo peculiar changes concerning the formation of their bases. Some others take certain substitutes in the conjugational tenses and moods before the conjugational affix is added.

The penultimate vowel of many roots is lengthened.

a→ā :- *kram→krām*[64]; *klam→klām;ācam→ācām*[65]

i→ī :- *ṣṭhiv→ṣṭhīv*[66]

u→ū :- *guha→gūha*[67]

Where the penultimate *r* or *v* of a root is followed by a consonant, the vowels *i-u-ṛ-ḷ* preceding the *r* or *v* are lengthened.[68]

Examples: *hurcch→hūrcch, murcch→ mūrcch*

---

[64] *kramaḥ parasmaipadeṣu*; hence not in Atmanepada
[65] *cam* alone does not lengthens its vowel
[66] *ṣṭhivuklamucamāmśiti*
[67] *ūdūpadhāyāgohaḥ*
[68] *upadhāyāñca* 8/2/78

The vowel of *mṛj* takes vṛddhi[69]; *kḷp* and *kṛp* becomes *kalp*[70] and the *s* of *lasj* and *sasj* is changed to *j*.[71]

Illustrations: *mṛj →mārj; kḷp →kalp; kṛp →kalp;lasj →lajj;sasj →sajj.*

vṛddhi of *a* is *ā*, *i/ī* is *e*, *u/ū* is *au*, *ṛ/ṝ* is *ār*, *ḷ* is *āl.*

The following roots have *āy* added on to them in the conjugational tenses and moods.

Examples: *gup →gopāy; dhūp →dhūpāy; vicch →vicchāy; paṇ →paṇāy; pan →panāy.*[72]

The root *kam* lengthens its vowel and has *ay* added on to it before the conjugational sign.

kam→kāmay[73]

Root *ṛt* has *īy* added on to it before the conjugational sign; *ṛt→ṛtīy*[74]

Many roots drop their nasals: *rañj→raj*[75]*; daṁś →daś; sañj →saj; svañj →svaj*[76]

jabh adopts a nasal, *jabh→jambh*[77]

The following roots take the substitutes before the conjugational sign is applied:

pā→pib; ghrā→jighr ; dhmā →dhm ; sthā →tiṣṭh; mnā →man; dā →yacch;

dṛś →paśy; ṛ →ṛcch; sṛ →dhāv; śad →śīy; sad → sīd; gam → gacch; yam →yacch.[78]

The root *śad,* though parasmai, becomes *ātmanepadi* in conjugational tenses and moods.

The following seven roots, though being desiderative in their forms, are considered as primitive ones. These retain the same base in the conjugational as well as in the non-conjugational tenses and moods.

kit(cikits), badh (bībhats), dāt(dīdāṁs), śān(śīśāṁs), mān(mīmāṁs), tij (titikṣ), gup (jugups)

---

[69] *mṛjervṛddhiḥ* 7/2/114, *uraṇraparaḷ* 1/1/51

[70] *kṛpo ro laḥ* 8/2/18

[71] *stoḥ ścunāścuḥ* 8/4/40; *jhalāṁ jaś jhaśi* 8/4/53

[72] *gupūdhūpavicchipaṇipanibhya āyaḥ* 3/1/28

[73] *kamerṇiṅ* 3/1/10

[74] *ṛterīyaṅ* 3/1/29

[75] *rañjeśca* 6/4/26

[76] *daṁśasañjasyañjośapi* 6/4/25

[77] *radhijabhoraci* 7/1/61

[78] *pāghrādhmāsthāmnādāṇdṛśyartisartiśadasadāmpibajighradhamatiṣṭhamanayacchapaśyarchadhauśīyasīdāḥ* 7/3/78

Following roots are declined like the 5th conjugation:
śru[79] *(śṛṇu); dhinv (dhinu); kṛṇv*[80] *(kṛṇu)*

Some roots add a penultimate nasal necessarily in the conjugational moods and tenses:
Parasmai: *kad (kand); cad (cand); bid (bind); bhid (bhind).*
Ātmanepada: *ah (aṁh); piḍra (piṇḍra); śuṭh (śuṇṭh)*

The Imperfect Tense (laṅ):

The roots are prefixed with augment *a* in this tense.
This *a* is replaced by *ā* if the root begins with a vowel.
When this *a* is followed by *i-u-ṛ* short or long becomes *ai-au-ār* respectively.
In case a root is prefixed with an *upasarga*, the augment *a* or *ā* comes between the *upasarga* and the root, and the sandhi rules have to be applied:
Similar vowels, followed by similar ones, substitute for both the same vowel lengthened.
When a dissimilar vowel follows *i, u, ṛ, ḷ,* short or long, *y, v, r, l* are respectively substituted for them.

The Imperative Mood (loṭ):

Imperative mood also follows the same process. However, in the first person singular of *parasmaipada*, if *n* follows *ṛ-r* or *ṣ* in the same word or follows any vowels, semi-vowels (*l* expected), and letters of the gluttural or labial class with *ṛ-r* or *ṣ* before them, it is changed to *ṇ*. If this *n* is at the end of a grammatical form, as in *aharan*, or is followed by a letter of the dental class, it remains changed.

Potential Mood (vidhiliṅ):

No major changes occur in the base formation in this tense. The formation of base is, for the most part, similar to that of present tense.

Perfect Tense (liṭ):

---

[79] *śruvaḥ śṛ ca* 3/1/74
[80] *dhinvikṛṇvyoraca* 3/1/80; *atolopaḥ* 6/4/48

The perfect tense is of two kinds: Reduplicated Perfect and Periphrastic Perfect. Some roots adopt exclusively the former and some the latter. Also, there are other roots which take either of them.

The Reduplicative Perfect: All primitive or monosyllabic roots beginning with *a, ā,* or *i, u, ṛ* or with any consonant and the roots *ṛcch,* and *ūrṇu* take the reduplicative perfect.

The Periphrastic Perfect: All primitive roots of more than one syllable except *ūrṇu*,[81] all roots beginning with long *ī, ū, ṝ* or prosodically long vowel, *i.e.* a short vowel followed by a conjunct consonant except *ṛcch*,[82] roots beginning with *e* and *o*, all derived roots such as those of the 10th conjugation, causals, desideratives and others, and the roots *day-ay-kās* and *ās*[83] take the periphrastic perfect.

The roots *uṣ, vid, jāgṛ, bhī, hī, bhṛ, hu* and *daridrā* admits of both the perfects.[84]

Three terminations *tha, va, ma* in the parasmaipada, and four *vahe, mahe, se* and *dhve* in the *ātmanepada* may adopt the augment *i*.

Rules regarding augment *i* in the perfect are as follows:

1. All *seṭ* roots admit *i* necessarily.
2. All *veṭ* roots adopt *i* optionally.
3. *Aniṭ* roots *kṛ-sṛ-bhṛ-vṛ-stu-dru-sru-śru* do not admit *i* in the perfect tense.[85] Root *vṛ*, however, takes *i* optionally before *tha*. Root *kṛ* with *sam-pari* or *upa* is also an exception.
4. *Aniṭ* roots ending in short *ṛ* admit *i* necessarily before *va-ma* or *vahe-mahe-se* and *dhve* but not before *tha*. Root *ṛ* (1 and 3 conj.) parasmaipada is an exception of this rule.
5. *Aniṭ* roots with a final vowel except short *ṛ* or having an *a* in them take *i* optionally before *tha* and necessarily before *va-ma* or *vahe-mahe-se* and *dhve*.
6. All other *aniṭ* roots take *i* necessarily in the perfect.

---

[81] ūrṇoterām netivācyam
[82] ijādergurumato'nṛcchaḥ 3/1/36
[83] dayāyāsaśca 3/1/37
[84] uṣavidajāgṛbhyo'nyatarasyām 3/1/53; bhī-hrībhṛhuvāṁśluvacca 3/1/39
[85] *kṛsṛbhṛvṛstudrusruśruvoliṭi* 7/2/13

When the 2nd person plural terminations *ṣīdhvam, dhvam* and *dhve* of the benedictive, aorist and perfect follow a vowel except *a* or *ā* or follow *y-v-r-l* or *h*, the *dh* is changed to *ḍh* necessarily in the absence of the intermediate *i* is preceded y any vowel except a or ā or the consonants *y-v-r-l* or *h*.

The root is reduplicated according to the following rules of reduplication.[86]

Rules of Reduplication:

Roots consisting of one vowel reduplicate, i.e. repeat or double the initial vowel, if any, or the initial consonant together with the following vowel.[87] Thus *ikh* becomes *īkh* and *puṣ* becomes *pupuṣ*.

In the case of roots consisting of two or more vowels the consonant following the initial vowel is reduplicated instead of the initial vowel.[88]

Roots with an initial vowel and *n, d, r* as the first member of a conjunct consonant following it, reduplicate the second member of a conjunct consonant.[89] Thus *ūrṇu* becomes *ūrṇunu*, *n* being repeated instead of *r*.

Roots with an initial conjunct consonant reduplicate the first member of it only with *rh* following vowel.[90] Thus *druh* become *dudruh*.

Roots with an initial conjunct,the first member of which is a sibilant and the second a hard consonant, reduplicate the second member with the following vowel.[91] Thus *spardh* becomes *paspardh*.The syllable so reduplicated or repeated is called the *reduplicative*

---

[86] *liṭidhātoranabhyāsasya* 6/1/8
[87] *ekācodve prathamasya* 6/1/1
[88] *ajāderdvitīyasya* 6/1/2
[89] *nandrāḥ saṁyogādayaḥ* 6/1/3
[90] *halādiḥ śeṣaḥ* 7/4/60
[91] *śarpūrvāḥ khayaḥ* 7/4/6

syllable.[92] Thus the first *i* in *ikh*, *pu* in *pupuṣ*, and *du* in *dudruh* is the reduplicative syllable.

In the reduplicative syllable, a radical hard or soft aspirate i.e. the second or fourth letter of a class is replaced by its corresponding hard or soft unaspirate, i.e. the first or third letter of the class.[93] Thus sphuṭ becomes *phusphuṭ* and then *pusphuṭ;* dhinv, *phaṇ-bhaj* becomes *didhinv, paphaṇ* and *babhaj*.

A reduplicative long vowel becomes short, and *ṛ* is replaced by *a*.

bādh → bābādh → babādh; bhāṣ → babhāṣ; bhṛ →bhṛ-bhṛ → babhṛ

The parasmaipada singulars are strong; all other terminations are weak.

The penultimate short of roots take guṇa before the strong terminations and the final vowel and penultimate a take vṛddhi necessarily before the a of the third person singular and optionally before the a of the first person singular parasmaipada. Before the tha of the 2nd person singular parasmai, the final vowel of the root takes guṇa and penultimate a remains the same.

Irregularities regarding guṇa and vṛddhi.

1. the vowel of *vid* is not gunated before the affix *ām* which is strong.
2. the *ṛ* of *jāgṛ* takes guṇa before any *sārvadhātuka* or *ārdhadhātuka* affix except *vi*, *ciṇ* (*i*), *ṇal* (*a* 1st and 3rd person singular perfect) or one distinguished by a mute *ṅ*.

The Aorist (luṅ)

The personal terminations in this tense undergo certain peculiarities and it may, therefore, be classified into the Radical Aorist and the Sibilant Aorist. In the Radical Aorist, the terminations are added immediately to the root whereas *s* or *siṣ* or *sa* is prefixed to the terminations in the Sibilant Aorist, before fixing the terminations to the root.

There are seven varieties of terminations in the Aorist. They have been illustrated in the next chapter in detail.

---

[92] *pūrvo'bhyāsaḥ* 6/2/4

[93] *abhyāse carca* 8/4/54

The simple future (lṛṭ)

The final vowel and the penultimate short of roots take guna.

The ending *s* of a root, when followed by any termination, beginning with *s* and belonging to the non-conjugational tenses and moods, takes *t* as its substitute.

Aniṭ roots, having *ṛ* for their penultimate change it to *ra* optionally before a strong termination beginning with any consonant except a nasal or a semi-vowel. Thus *sṛj* and *dṛś* become *sraj* and *draś*.

The intermediate *i* is optionally lengthened in the case of *vṛ* and roots ending in long *ṝ* before the terminations of the non-conjugational tenses and moods except those of the perfect. It is necessarily lengthened in the case of the root *grah* under the same circumstances.

Aniṭ roots: Aniṭ roots don't take *i*.

Exception:

1. Roots with a final *ṛ* and the roots han admit *i* in the future and conditional.
2. *gam* in the parasmai takes *i* in the future, conditional and desiderative, gam (substitute for *i* [2nd parasmai], and *i* with *adhi* 'to remember') does the same in the desiderative.
3. The root *daridrā* drops its *ā*.

Seṭ roots: Seṭ roots admit *i*.

Exceptions:

1. The roots *vṛt, vṛdh* and *śṛdh*, though ātmanepadi, become optionally parasmaipada in the future, conditional, and desiderative, in which case they do not take *i* before the parasmaipada terminations.

Veṭ roots: Veṭ roots take *i* optionally.

However, roots *syand* and *kḷp* follow exception.

The periphrastic future (luṭ)

Seṭ, veṭ and *aniṭ* roots adopt the augment ***i*** according to the nature of the root.

Exception: The roots *iṣ, sah, lubh, ruṣ and riṣ,* though seṭ, take *i* optionally when followed by any non-conjugational termination beginning with *t*.

The final vowel and the penultimate short of roots take their guna substitute before the terminations are applied.

The root *kḷp* is optionally parasmaipada in the two futures, conditional and desiderative, in which case it does not take *i* before the parasmaipada terminations.

The conditional mood (lṛṅ)

a is prefixed to the roots. Roots like *vṛt, vṛdh, śṛdh, syand, kḷp, kṛt, cṛt, tṛd* etc. follow the same rules that of simple future.

The benedictive mood

The *dh* of *sīdhvam* is to be changed to *ḍh* if they follow a vowel other than *a* or *ā*; or follow *y, v, r, l, h.* The change is necessary in the absence of the intermediate *i* and is optional if intermediate *i* is preceded by any vowel except *a* or *ā*; or *y,v,r,l,h.*

The parasmai terminations are weak except in the case of roots ending in short *ṛ* which is preceded by a conjunct consonant and the roots *jāgṛ* and *ṛ.*

The ātmanepada terminations are the strong except in the case of *aniṭ* roots and roots ending in *ṛ* short or long when the terminations do not take *i;* viz. the ending *ṛ* vowel (short or long) of roots whether *seṭ, veṭ* or *aniṭ* when they admit *i* and the final vowels and the penultimate short of roots of all *seṭ* roots except *gur* (4th Atmane), the penultimate vowel of *veṭ* roots *kḷp* and *gṛh* when they take *i* and the final of *sū* and *dhū* with or without *i* and the ending *i-ī* and *u* of *aniṭ* roots take *guṇa* in the ātmanepada benedictive.

The parasmai terminations of the benedictive mood begin with *y*. Consequently, *seṭ* roots do not take the augment *i*. Hence all roots whether *seṭ, veṭ* or *aniṭ* do not take the augment *i* in the benedictive parasmai.

Roots ending in vowels: Roots ending in *ā-e-ai* and *o* and not preceded by a conjunct consonant retain the final ā before the terminations.

Exceptions:

1. Roots ending in *ā, e, ai* and *o* and preceded by a conjunct consonant change the final *ā* to *e* optionally. The root *daridrā* drops its *ā*.
2. The *ghu* verbs *dā, dhā , dā , do, de, dhe* and the roots *mā-sthā-gai-pā-hā* and *so* change the ending *ā* to *e* necessarily.
3. The roots *ve-vye-hve* and *jyā* take *samprasāraṇa* and lengthen the vowel in the benedictive and before the *ya* of the passive; viz. *ve* and *hve* substitute *ū* for *ve*, while *jye* and *jyā* substitute *ī* for *ye* and *yā*. The base, therefore, in each case becomes *ū, hū, vī* and *jī*.
4. The ending *i* and *u* of roots lengthen before the weak terminations of the benedictive parasmai, and the *ya* of the passive, as *cī* and *dhū* from *ci* and *dhu*.

Exception: The root *śvi* takes samprasāraṇa and lengthens the vowel in the benedictive, and before the *ya* of the passive, the base becomes *śū*.

Roots ending in *ṛ* change to *ri*. However if the *ṛ* is preceded by a conjunct consonant, it takes *guṇa* before the weak terminations of the benedictive parasmai and the *ya* of the passive; as *kri* from *kṛ*, and *smar* from *smṛ*. The roots *jāgṛ* and *ṛ* (1,3 parasmai) take *guṇa*.

Roots ending in *ṝ* change it to *īr* but if *ṝ* follows a labial or *v*, it is changed to *ūr* before the weak terminations of the benedictive parasmai, and the *ya* of the passive; as *stīr* and vūr from *stṝ* and *vṝ*.

Roots ending in consonants:

The roots *vac, svap, yaj, vap, vah, vas, vad, grah, vyadh, vaś, vyac, vraśc, pracch, bhrasj* take *samprasāraṇa* in the *parasmaipada* only and before the *ya* of the passive. The roots *div, siv, ṣṭhiv* and *sriv* lengthen their penultimate *i* before the terminations of the benedictive and before the *ya* of the passive.

The roots *jan, san* and *khan* optionally substitute *ā* for the final *n* and the root *ūh* when with a preposition shortens its penultimate before the parasmai terminations of the benedictive and the *ya* of the passive.

The *ay* of the 10th conj. and causals and the denominative *y* are dropped in the benedictive parasmaipada but retained in the ātmanepada.

The penultimate nasal of the following parasmaipada roots is dropped in the benedictive and before the *ya* of the passive:

añc, vañc, kuñc, luñc, sañj, skand, tump, daṁś and *śaṁs, rañj, manth.*

Exception:

1. The root *añc* meaning to worship does not drop its nasal.
2. The penultimate nasal of all *ātmanepada* roots and that of the following parasmaipada ones is invariably retained in the benedictive and before the *ya* of the passive:

   taṅk and *kāṅkṣ, liṅg, laṅgh, daṁh, khañj, guñj, luṇṭ, kuṇṭh, luṇṭh, maṇḍ, muṇḍ, krand, kland, klind, nind, nand, manth* and *cand, dhinv* etc.

   The root *jabh* (1st ātmane) adopts a nasal in the benedictive and in the passive, as *jambhiṣīya, jambhiṣīvahi, jambhyate.*

Ātmanepada:

Seṭ roots take *i* necessarily.

Exception: Roots ending in long *ṛ* take *i* optionally in the ātmanepada benedictive. This intermediate i is not lengthened in the benedictive as in the perfect.

In the case of Seṭ roots, final vowels, penultimate short of roots and the ending long *ṛ* of roots, when they take *i*, take their guṇa substitute in the ātmanepada benedictive.

When roots ending in long *ṛ* do not take *i*, it is changed to *īr*, and to *ūr* if a labial or *v* precedes.

aniṭ roots do not take *i*.

Exceptions: Roots ending in short *ṛ* preceded by a conjunct consonant admit *i* optionally.

Final *i-ī-u* and final *ṛ* preceded by a conjunct consonant when it admits *i* take their guṇa substitute while roots ending *ā-e-ai-o;* roots ending in *ṛ* not preceded by a conjunct

consonant and also in *ṛ* preceded by a conjunct consonant but when it does not take *i* remains unchanged. *kṛ* and *sīṛ* are the examples.

veṭ roots take *i* optionally.

All *seṭ* roots ending in long *ṛ* and the root *vṛ;* all aniṭ roots ending in short *ṛ* preceded by a conjunct consonant and the veṭ roots *klp, gṛh, guh, svṛ* take *i* optionally in the *ātmanepada* benedictive.

The roots *sū* and *dhū* take guṇa with or without *i*. The roots *vṛ, klp, gṛh, svṛ* take guṇa when with *i* and remain unchanged when without *i*; while the root *guh* lengthens its vowel instead of taking guṇa when it takes *i* and remains unchanged when without *i*.

The rules stated above are illustrated with examples of roots bhū (भू) and edha (एध) of bhvādigaṇa. The former is parasmaipadi and the latter is ātmanepadi. They may be taken as samples representing the forms of whole class.

<u>भू कर्तृवाच्य active</u>

laṭ भवति भवतः भवन्ति भवसि भवथः भवथ भवामि भवावः भवामः

liṭ बभूव बभूवतुः बभूवुः बभूविथ बभूवथुः बभूव बभूव बभूविव बभूविम

luṭ भविता भवितारौ भवितारः भवितासि भवितास्थः भवितास्थ भवितास्मि भवितास्वः भवितास्मः

lṛṭ भविष्यति भविष्यतः भविष्यन्ति भविष्यसि भविष्यथः भविष्यथ भविष्यामि भविष्यावः भविष्यामः

loṭ भवतु,तात् भवताम् भवन्तु भव,तात् भवतम् भवत भवानि भवाव भवाम

laṅ अभवत् अभवताम् अभवन् अभवः अभवतम् अभवत अभवम् अभवाव अभवाम

Vliṅ भवेत् भवेताम् भवेयुः भवेः भवेतम् भवेत भवेयम् भवेव भवेम

Aliṅ भूयात् भूयास्ताम् भूयासुः भूयाः भूयास्तम् भूयास्त भूयासम् भूयास्व भूयास्म

luṅ अभूत् अभूताम् अभूवन् अभूः अभूतम् अभूत अभूवम् अभूव अभूम

lṛṅ अभविष्यत् अभविष्यताम् अभविष्यन् अभविष्यः अभविष्यतम् अभविष्यत अभविष्यम् अभविष्याव अभविष्याम

एध कर्तृवाच्य active

laṭ एधते एधेते एधन्ते एधसे एधेथे एधध्वे एधे एधावहे एधामहे

liṭ एधाञ्चक्रे एधाञ्चक्राते एधाञ्चक्रिरे एधाञ्चकृषे एधाञ्चक्राथे एधाञ्चकृढ्वे एधाञ्चक्रे एधाञ्चकृवहे एधाञ्चकृमहे

(optional) एधाम्बभूवे एधाम्बभूवाते एधाम्बभूविरे एधाम्बभूविषे एधाम्बभूवाथे एधाम्बभूविढ्वे,ध्वे एधाम्बभूवे एधाम्बभूविवहे एधाम्बभूविमहे

luṭ एधिता एधितारौ एधितारः एधितासे एधितासाथे एधिताध्वे एधिताहे एधितास्वहे एधितास्महे

lṛṭ एधिष्यते एधिष्येते एधिष्यन्ते एधिष्यसे एधिष्येथे एधिष्यध्वे एधिष्ये एधिष्यावहे एधिष्यामहे

loṭ एधताम् एधेताम् एधन्ताम् एधस्व एधेथाम् एधध्वम् एधै एधावहै एधामहै

laṅ ऐधत ऐधेताम् ऐधन्त ऐधथाः ऐधेथाम् ऐधध्वम् ऐधे ऐधावहि ऐधामहि

Vliṅ एधेत एधेयाताम् एधेरन् एधेथाः एधेयाथाम् एधेध्वम् एधेय एधेवहि एधेमहि

Aliṅ एधिषीष्ट एधिषीयास्ताम् एधिषीरन् एधिषीष्ठाः एधिषीयास्थाम् एधिषीध्वम् एधिषीय एधिषीवहि एधिषीमहि

luṅ ऐधिष्ट ऐधिषाताम् ऐधिषत ऐधिष्ठाः ऐधिषाथाम् ऐधिढ्वम् ऐधिषि ऐधिष्वहि ऐधिष्महि

lṛṅ ऐधिष्यत ऐधिष्येताम् ऐधिष्यन्त ऐधिष्यथाः ऐधिष्येथाम् ऐधिष्यध्वम् ऐधिष्ये ऐधिष्यावहि ऐधिष्यामहि

2.2 .2 The passive voice and the impersonal constructions:

The passive and the impersonal forms are formed in one and the same manner from all roots of ten classes without any conjugational distinction. They, however, differ in their

formation in the conjugational and non-conjugational tenses and moods. The roots also undergo certain peculiar modifications before they take the terminations.

Conjugational tenses and moods:

The passive and the impersonal forms in the four conjugational tenses and moods, whether the root is parasmaipada or ātmanepadi in the active, are formed by adding *ya* to the original root and then applying the ātmanepada terminations,.

This *ya* is weak and no guṇa or *vṛddhi* substitute takes place before it. Exceptionally, roots of the 10th conjugation capable of taking *guṇa* or *vṛddhi* substitute in the active voice, do the same in the passive also.

Some roots such as *dā, de, dhā, gai, sthā, pā* change the ending *ā* to *ī* before the *ya* of passive. The ending *ā* of the other roots and the *ā* to be substituted in the case of the roots that end with *e, o* and *ai,* remain unchanged.

Roots ending in short *i* and *u*, in *ṛ* preceded by a simple or conjunct consonant and roots ending in long *ṝ* take the same changes which they undergo before the parasmai terminations of the benedictive. Roots *ṛ, jāgṛ* take their *guṇa.*

Roots ending in consonants such as *vac, svap, yaj, vap, vah, vas, vad, grah, vyadh, vaś, vyac, vraśc, pracch, bhrasj; vañc, kuñc, luñc, sañj, skand, tump, damś, śaṁs, rañj, manth,* etc. take *samprasāraṇa* and drop their nasal before the *ya* of the passive also. Root *khan* optionally substitutes *ā* for its final, *khan→khanyate, khāyate*. Of Ātmanepadi roots having a penultimate nasal, *syand, sraṁs, srambh* and *khanj* drop their nasal before the *ya* of the passive.

Non Conjugational tenses and moods:

The perfect:

The passive of the reduplicated perfect is formed by affixing the ātmanepadi terminations to every root whether it is parasmaipadi or ātmanepadi in the active, as *gam* → *jagme, yaj* →*ije, bhid* → *bibhide, añj* → *ānaje.*

The periphrastic perfect of the passive is formed by affixing *ām* to the root, and then appending the ātmanepada forms of three auxiliary verbs *am, kṛ* and *bhū* whether the root is parasmai or ātmane. *kath* → *kathayāñcakre, kathayāmbabhūve, kathayāmāse.*

The Aorist:

The first and the sixth varieties are exclusively *parasmaipadi.* All the roots belonging to them are *aniṭ* except the roots bhū which is *seṭ. bhū* takes *ātmanepada* forms of the 5th variety in the passive and impersonal of the aorist, and all the rest takes terminations of 4th variety.

The ātmanepada forms of the 2nd and 3rd varieties are not used in the passive and the impersonal of the aorist. Consequently roots belonging to them take the *ātmanepadi* forms of the fourth, fifth or seventh varieties in the passive and the impersonal of the aorist according as they are *seṭ* or *aniṭ.* The passive and impersonal forms of the 4th, 5th and 7th varieties of the aorist are simply formed by appending the *ātmanepada* terminations to the prepared base.

The third person singular of the aorist passive and impersonal of all roots except *tap* is made up by prefixing to the root the augment and adding to it the termination *i* instead of *sta* or *sata.* The root *tap*, in the sense of repenting and in the reciprocal passive too, does not take *i.*

Before the passive *i* of the third person singular of the aorist, the roots take some changes:

The penultimate short of the root takes its guṇa before the passive *i.*

Seṭ roots ending in *m* and the roots *vadh* and *jan* do not lengthen the penultimate vowel before the passive *i.*

The passive and impersonal forms of the aorist of roots ending in a vowel, of roots ending in *ay*, and of roots *han, grah* and *dṛśare* optionally formed by adding *i* to the root even when it is *aniṭ* except in the third person singular.

The two futures, the conditional and the benedictive:

The passive and the impersonal forms of the two futures, the conditional and the benedictive are formed simply by adding the *ātmane* terminations to the prepared base, whether the root is *parasmai* or *ātmane* in the active. However, the passive and impersonal forms in the same *lakāras* of roots ending in vowels and of roots *grah, dṛś* and *han* may optionally be formed by adding *i* to the root, though it is *aniṭ.*

भू कर्मवाच्य (passive)

laṭ	भूयते	भूयेते	भूयन्ते	भूयसे	भूयेथेभूयध्वे	भूये	भूयावहे	भूयामहे
liṭ	बभूवे	बभूवाते	बभूविरे	बभूविषे	बभूवाथे	बभूविढ्वे,ध्वे	बभूवे	बभूविवहे बभूविमहे
luṭ	भविता,भाविता	भवितारौ	भवितारः	भवितासे	भवितासाथेभविताध्वे	भविताहे	भवितास्वहे भवितास्महे	
lṛṭ	भविष्यते	भविष्येते	भविष्यन्ते	भविष्यसे	भविष्येथे	भविष्यध्वे	भविष्ये	भविष्यावहे भविष्यामहे
loṭ	भूयताम्	भूयेताम्	भूयन्ताम्	भूयस्व	भूयेथाम्	भूयध्वम्	भुयै	भूयावहै भूयामहै
laṅ	अभूयत	अभूयेताम्	अभूयन्त	अभूयथाः	अभूयेथाम्	अभूयध्वम्	अभूये	अभूयावहि अभूयामहि
Vliṅ	भूयेत	भूयेयाताम्	भूयेरन्	भूयेथाः	भूयेयाथाम्	भूयेध्वम्	भूयेय	भूयेवहि भूयेमहि
Aliṅ	अभविषीष्ट,भावि	भविषीयास्ताम्	भविषीरन्	भविषीष्ठाः	भविषीयास्थाम्	भविषीध्वम्	भविषीय	भविषीवहि भविषीमहि
luṅ	अभावि	अभाविषाताम्,अभवि	अभाविषत	अभाविष्ठाः	अभाविषाथाम्	अभाविध्वम्	अभाविषि	अबोभूयिष्वहि अबोभूयिष्महि
lṛṅ	अभविष्यत	अभविष्येताम्	अभविष्यन्त	अभविष्यथाः	अभविष्येथाम्	अभविष्यध्वम्	अभविष्ये	अभविष्यावहि अभविष्यामहि

एध कर्मवाच्य (passive)

laṭ	एध्यते	एध्येते	एध्यन्ते	एध्यसे	एध्येथे	एध्यध्वे	एध्ये	एध्यावहे	एध्यामहे
luṭ	एधिता	एधितारौ	एधितारः	एधितासे	एधितासाथे	एधिताध्वे	एधिताहे	एधितास्वहे	एधितास्महे
lṛṭ	एधिष्यते	एधिष्येते	एधिष्यन्ते	एधिष्यसे	एधिष्येथे	एधिष्यध्वे	एधिष्ये	एधिष्यावहे	एधिष्यामहे
loṭ	भूयताम्	भूयेताम्	भूयन्ताम्	भूयस्व	भूयेथाम्	भूयध्वम्	भुयै	भूयावहै	भूयामहै
laṅ	ऐधयत	ऐधयेताम्	ऐधयन्त	ऐधयथाः	ऐधयेथाम्	ऐधयध्वम्	ऐधये	ऐधयावहि	ऐधयामहि
Vliṅ	एध्येत	एध्येयाताम्	एध्येरन्	भूयेथाः	भूयेयाथाम्	भूयेध्वम्	भूयेय	भूयेवहि	भूयेमहि
Aliṅ	एधिषीष्ट,भावि	एधिषीयास्ताम्	एधिषीरन्	एधिषीष्ठाः	एधिषीयास्थाम्	एधिषीध्वम्	एधिषीय	एधिषीवहि	एधिषीमहि
luṅ	ऐधि	ऐधिषाताम्	ऐधिषत	ऐधिष्ठाः	ऐधिषाथाम्				
lṛṅ	ऐधिष्यत	ऐधिष्येताम्	ऐधिष्यन्त	ऐधिष्यथाः	ऐधिष्येथाम्	ऐधिष्यध्वम्	ऐधिष्ये	ऐधिष्यावहि	ऐधिष्यामहि

2.2 .3 Causals:

Roots ending in *ā, e, ai* and *o*; roots *ṛ*, *knūy*, *kṣmāy, ji* add on the augment *p* before aya. The *y* of *knūy* and *kṣmāy* is dropped by the rule that *v* or *y* is dropped before any consonant except *y*. The roots ***kṣai***, *śrā* or *śrai* and *jñā* shorten their *ā* necessarily before p and *glai* and *srā* do it optionally when not preceded by a preposition. The root *smi* is regular as *smāyayati*.

The roots *śo, cho, so, hve, vye, ve, sai* and *pā* take the augment *y* instead of *p* before *aya*. Root *vai*, when meaning to shake or tremble take the augment *j* before *aya*. The same root, however, in the other senses also form their causals regularly.
The final vowels take *vṛddhi* substitute before *aya*, as *kṣāyaya* from *kṣi*.

Exception: *smṛ* in the sense of 'to cause', 'to long for', 'to regret' and the roots *ṛ*, *hrī*, *rī* take *guṇa* substitute instead of *vṛddhi*.

ā is necessarily substituted for the vowels of *krī, i, ji,mi mī* to destroy *dī* and for those of *smi* if the effect is produced by the agent, *sidh* if the import is not a spiritual topic. ā is optionally substituted for the vowels of *ci* and *sphur* and for those of *bhī* if the effect is produced by the agent, and *dī* if it means 'to cause', and 'to conceive'.
The penultimate *a* is changed to its *vṛddhi* substitute before aya, as *khānaya* from *khan*; *chādaya* from *chad*.
Exception-1: Roots ending in *am* (except *kam, am, yam*) and the following roots which are marked as *mit* in the *dhātupāṭha* do not change their vowel:-
ak, cak 'to be satisfied', *stak, kagv, ag, kag, rag, lag, sag, stag, hag.* and *hlag; ghaṭ, bhaṭ, naṭ, gaḍ, laḍ* (*laḍayati/lāḍayati* in different senses), *heḍ* (*hiḍayati/heḍyati* in different senses), *kaṇ, raṇ,phaṇ, caṇ, śraṇ, śaṇ, vyath, prath, śrath, knath, krath, klath, mrad, skhad, van, dhvan, svan, krap, ram, tvar, jvar, dakṣ, pras* etc.

Exception-2: The roots *van, vam nam, jval, hval, hmal, dhvan* and *cal* lengthen their *a* optionally when not preceded by a preposition.
The roots *rabh, labh, jabh* insert a nasal before their final. *lambhayati/te*
Penultimate short vowel takes *guṇa* substitute before *aya*, as *tepaya* from *tip*.
sphāy changes its *y* to *v* in the causal, and *bhrasj* assumes two forms *bharj* and *bhrajj*.

Conjugations:
Causal roots are generally conjugated in both the padas.

Exception: The causals forms of roots having the sense of 'to swallow' or 'to move' and of the following roots are conjugated in the parasmaipada only: *īrṣy, garv, jvar, nand, lal, klind, galbh, jan, jabh, ram, vyath, klam, ṣṭhiv, bhram.*

The following roots have the sense of to swallow or to move:
kṣar, khād, khel, gal, cam, cal, dru, sru, vel, vep, kamp, gras, ceṣṭ, pru.

भू णिजन्त (Causal) कर्तृवाच्य (active) परस्मैपद

laṭ भावयति भावयतः भावयन्ति

liṭ भावयाञ्चकार भावयाञ्चक्रतुः भावयाञ्चक्रुः भावयाञ्चकर्थ भावयाञ्चक्रथुः
भावयाञ्चक्र भावयाञ्चकार भावयाञ्चकृव भावयाञ्चकृम

luṭ भावयिता भावयितारौ भावयितारः

lṛṭ भावयिष्यति भावयिष्यतः भावयिष्यन्ति

loṭ भावयतु,तात् भावयताम् भावयन्तु

laṅ अभावयत् अभावयताम् अभावयन्

Vliṅ भावयेत् भावयेताम् भावयेयुः

Aliṅ भाव्यात् भाव्यास्ताम् भाव्यासुः भाव्याः भाव्यास्तम्

luṅ अबीभवत् अबीभवताम् अबीभवन् अबीभवः अबीभवतम्

lṛṅ अभावयिष्यत् अभावयिष्यताम् अभावयिष्यन्

भू णिजन्त (Causal) कर्तृवाच्य (active) आत्मनेपद

laṭ भावयते भावयेते भावयन्ते

liṭ भावयाञ्चक्रे भावयाञ्चक्राते भावयाञ्चक्रिरे भावयाञ्चकृषे भावयाञ्चक्राथे भावयाञ्चकृध्वे
भावयाञ्चक्रे भावयाञ्चकृवहे भावयाञ्चकृमहे

luṭ भावयिता भावयितारौ भावयितारः भावयितासे

lṛṭ भावयिष्यते भावयिष्येते भावयिष्यन्ते

loṭ भावयताम् भावयेताम् भावयन्ताम्

laṅ अभावयत अभावयेताम्अभावयन्त

Vliṅ भावयेत भावयेयाताम् भावयेरन् भावयेथाः

Aliṅ भावयिषीष्ट भावयिषीयास्ताम् भावयिषीरन् भावयिषीष्ठाः

luṅ अबीभवत अबीभवेताम् अबीभवन्त अबीभवथाः अबीभवेथाम् अबीभवध्वम् अबीभवे अबीभवावहि अबीभवामहि

lṛṅ अभावयिष्यत अभावयिष्येताम् अभावयिष्यन्त

<u>एध णिजन्त (Causal) कर्तृवाच्य (active) परस्मैपद</u>

laṭ एधयति एधयतः एधयन्ति एधयसि एधयथः एधयथ एधयामि एधयावः एधयामः

liṭ एधयाञ्चकार एधयाञ्चक्रतुः एधयाञ्चक्रुः एधयाञ्चकर्थ एधयाञ्चक्रथुः एधयाञ्चक्र एधयाञ्चकार एधयाञ्चकृव एधयाञ्चकृम

luṭ एधयिता एधयितारौ एधयितारः एधयितासि

lṛṭ एधयिष्यति एधयिष्यतः एधयिष्यन्ति

loṭ एधयतु,तात् एधयताम् एधयन्तु एधय,तात्

laṅ ऐधयत् ऐधयताम् ऐधयन् ऐधयः ऐधयतम्

V.liṅ एधयेत् एधयेताम् एधयेयुः एधयेः एधयेतम्

A.liṅ एध्यात् एध्यास्ताम् एध्यासुः एध्याः एध्यास्तम्

luṅ ऐदिधत् ऐदिधताम् ऐदिधन् ऐदिधः ऐदिधतम् ऐदिधत ऐदिधम् ऐदिधाव ऐदिधाम

lṛṅ ऐधयिष्यत् ऐधयिष्यताम् ऐधयिष्यन् ऐधयिष्यः ऐधयिष्यतम् ऐधयिष्यत ऐधयिष्यम् ऐधयिष्याव ऐधयिष्याम

<u>णिजन्त Nijanta कर्तृवाच्य active आत्मनेपद</u>

laṭ एधयते एधयेते एधयन्ते एधयसे एधयेथे एधयध्वे एधये एधयावहे एधयामहे

liṭ एधयाञ्चक्रे एधयाञ्चक्राते एधयाञ्चक्रिरे एधयाञ्चकृषे

luṭ एधयिता एधयितारौ एधयितारः एधयितासे

lṛṭ एधयिष्यते एधयिष्येते एधयिष्यन्ते

loṭ एधयताम् एधयेताम् एधयन्ताम् एधयस्व

laṅ ऐधयत ऐधयेताम् ऐधयन्त ऐधयथाः

Vliṅ एधयेत एधयेयाताम् एधयेरन् एधयेथाः

Aliṅ एधिषीष्ट एधिषीयास्ताम् एधिषीरन् एधिषीष्ठाः एधिषीयास्थाम्

luṅ ऐदिधत ऐदिधेताम् ऐदिधन्त ऐदिधथाः ऐदिधेथाम् ऐदिधध्वम् ऐदिधे ऐदिधावहि ऐदिधामहि

lṛṅ ऐधयिष्यत ऐधयिष्येताम् ऐधयिष्यन्त ऐधयिष्यथाः

2.2 .4 Desideratives:

To form a desidrative base, the root is first reduplicated according to the general rules of reduplication. Then termination *s* is added on to it in the non-conjugational tenses and moods. In the conjugational tenses and moods, *a* is added to *s*.

Roots beginning with a vowel reduplicate the following consonant with *i* added on to it; or where necessary they first undergo the vowel changes and then take their reduplication, as *aṭiṭiṣati, īcikṣati, arciciṣati, arṇiniṣati, pratīṣiṣati, adhīṣiṣati* from the roots *aṭ, īkṣ, ṛ, arc, ṛṇ* etc.

The *a* of the reduplicative syllable is changed to *i*, and the characteristic *s* of the desiderative becomes *ṣ* , but the *s* of a primitive root remains unchanged when the desiderative *s* is changed to ṣ. When the latter remains unchanged the primitive *s* becomes ṣ; as *sinaniṣati-te* or *siṣāsati-te* from *san.* The root *stu* is an exception, it forms *tuṣṭūṣati.*other examples are *si, smi, su, sthā* and *sad.*

The desiderative base takes the same terminations of *parasmaipada* or *ātmanepada* or both as the primitive root does.

Exception:

The desiderative forms of *jñā, śru, smṛ,* and *dṛś* are conjugated in the *ātmanepada* while root *jñā* with prefix *anu* and root *śru* with *prati* or *ā* take parasmaipada terminations. Roots *klṛp, vṛt, vṛdh, śṛdh,* and *syand* may optionally form their desiderative in parasmai.

Peculiarities of *aniṭ, seṭ* and *veṭ* roots:

Aniṭ roots don't take *i* before *s*.

Excep: *smi, ṛ* take *i* necessarily before *s* as *sismayiṣate* and *aririṣati*.

The root *gam* takes *i* in the *parasmai* and rejects it in the *ātmanepada*, as *jigamiṣati, saṁjigaṁsate.*

Seṭ root takes *i* before *s*:

Exception: Roots ending in long *ū* and the following roots do not take *i* before *s*.

Roots ending in long *ṛ* and some other roots take *i* optionally before *s*.

pat → *pitsati, pipatiṣati*

The roots *vṛt, vṛdh* and *śṛdh* take *i* in the *ātmanepada* and reject it in the parasmai, but the root *kram* takes *i* in the parasmai and rejects it in the *ātmanepada*; as *vivartiṣate* or *vivṛtsati, vivardhiṣate* or *vivṛtsati*, *śiśardhiṣate* or *śiśṛtsati. cikramiṣate* or *cikraṁsate.*

Veṭ roots take *i* optionally before *s*.

Exception: The roots *guh-dhū* and *sū* do not take *jughukṣati-te, dudhūṣati-te, susūṣate.*

añj and *aś* take *i* necessarily as *añjijiṣate* and *aśiśiṣate*.

The roots *syand* and *kḷp* take *i* optionally in the *ātmanepada* and altogether reject it in the parasmai as *sisyandiṣate* or *sisyantsate*, *sisyantsati. cikalpiṣate* or *cikḷpsate, cikḷpsati.*

Roots ending in *iv* take *i* optionally before *s*, and this final *iv* is changed to *y* when the *s* is unaugmented. *kṣiv* → *cikṣeviṣati, cukṣyūṣati. ṣṭhiv* → *tiṣṭheviṣati, ṭiṣṭheviṣṭi, tuṣṭhyūṣati, ṭuṣṭhyūṣati.*

The desiderative characteristic *s* is considered weak when it does not take the argument *i*.

Special rules regarding radical vowel change in the desiderative are listed below:

The ending *i* and *u* of a root is lengthened and the ending *ṛ* and *ṝ* are changed to *īra,* or to *ūra* if a labial precedes when the *s* of the desiderative is weak or unaugmented; as *cikṣīṣati, bubhūrṣati-te* and *titīrṣati* from *kṣi, bhṛ* and *tṛṛ.*

When the *s* of the desiderative is strong, viz. augmented the final *i-u* and *ṛ* (short or long) take their guṇa substitute. *śvi-śiśvayiṣati, ḍī-ḍiḍayiṣate, pū-pipaviṣate* etc.

The penultimate short *ṛ* and *ḷ* of roots, and the penultimate short *i* and *u* of roots beginning with vowels take their guṇa substitute before the augmented s and remain unchanged before the unaugmented one. *uṣ-oṣiṣiṣati, vṛt-vivartiṣate, kḷp-cikalpiṣate.* Also *vivṛtsati* and *cikḷpsati* occur when the *s* is augmented.

But the penultimate short *i* and *u* of such roots as begin with consonants and end with any except *y* or *v* take *guṇa* optionally and the penultimate *i* of roots in *iv* does it necessarily when the *s* of the desiderative is strong, viz. augmented and the same vowels remain unchanged when it is unaugmented.

Roots ending in *e-ai* and *o* except *me,de,dhe* change their final vowel to *ā* before the *s* of the desiderative; as *kṣai-cikṣāsati, cho-cichchāsati.*

Irregularities regarding reduplication and vowel changes of certain roots in the desiderative, are furnished below:

Some roots reject reduplication and substitute short *i* for their vowel when the *s* is unaugmented; e.g. *rabh-ripsate, labh-lipsate, pat-pitsati, pipatiṣati.* Such a root, if ends in a vowel, add *t* to the desiderative *s. dā-ditsati, de-ditsate, me-mitsate* etc.

The roots *grah, svap, pracch, dyut* and *hve* take samprasāraṇa as *jighṛkṣati-te, suṣupsati, pipṛcchiṣati* and *didyutiṣate* or *didyotiṣate, juhūṣati-te.*

ji →jigīṣati

gup →jugopāyiṣati, jugopiṣati-te, jugupiṣati-te, jugupsati,

vicch →vivicchiṣati,vivicchāyiṣati

paṇ →pipaṇāyiṣati, pipaṇiṣate, kam-cikamiṣate/cikāmayiṣate

aj →ajijiṣati

mṛj →mimārjiṣati,mimṛkṣati

Conjugation of the desiderative base:

In the active voice, the desiderative base takes *a* and then it is conjugated like the base of a root of the 6th class. In the passive *ya* is added to the base and it is then conjugated like an ordinary Passive verb.

भू सन्नन्त sannanta कर्तृवाच्य active परस्मैपद

laṭ	बुभूषति	बुभूषतः	बुभूषन्ति			
liṭ	बुभूषाञ्चकार	बुभूषाञ्चक्रतुः	बुभूषाञ्चक्रुः	बुभूषाञ्चकर्थ	बुभूषाञ्चक्रथुः	
luṭ	बुभूषिता	बुभूषितारौ	बुभूषितारः			
lṛṭ	बुभूषिष्यति	बुभूषिष्यतः	बुभूषिष्यन्ति			
loṭ	बुभूषतु,तात्	बुभूषताम्	बुभूषन्तु			
laṅ	अबुभूषत्	अबुभूषताम्	अबुभूषन्			
Vliṅ	बुभूषेत्	बुभूषेताम्	बुभूषेयुः			
Aliṅ	बुभूष्यात्	बुभूष्यास्ताम्	बुभूष्यासुः			
luṅ	अबुभूषीत्	अबुभूषिष्टाम्	अबुभूषिषुः	अबुभूषीः	अबुभूषिष्टम्	
	अबुभूषिष्ट	अबुभूषिषम्	अबुभूषिष्व	अबुभूषिष्म		
lṛṅ	अबुभूषिष्यत्	अबुभूषिष्यताम्	अबुभूषिष्यन्			

भू सन्नन्त desiderative कर्तृवाच्य active आत्मनेपद

laṭ	बुभूषते	बुभूषेते	बुभूषन्ते			
liṭ	बुभूषाञ्चक्रे	बुभूषाञ्चक्राते	बुभूषाञ्चक्रिरे	बुभूषाञ्चकृषे	बुभूषाञ्चक्राथे	बुभूषाञ्चकृध्वे
	बुभूषाञ्चक्रे	बुभूषाञ्चकृवहे	बुभूषाञ्चकृमहे			
luṭ	बुभूषिता	बुभूषितारौ	बुभूषितारः	बुभूषितासे		
lṛṭ	बुभूषिष्यते	बुभूषिष्येते	बुभूषिष्यन्ते			
loṭ	बुभूष्ताम्	बुभूषेताम्	बुभूषन्ताम्			
laṅ	अबुभूषत	अबुभूषेताम्	अबुभूषन्त			

Vliṅ बुभूषेत बुभूषेयाताम् बुभूषेरन्

Aliṅ बुभूषिषीष्ट बुभूषिषीयास्ताम् बुभूषिषीरन बुभूषिषीष्ठाः बुभूषिषीयास्थाम्

luṅ

lṛṅ अबुभूषिष्यत अबुभूषिष्येताम् अबुभूषिष्यन्त

<u>एध सन्नन्त sannanta कर्तृवाच्य active आत्मनेपद</u>

laṭ एदिधिषते एदिधिषेते एदिधिषन्ते

liṭ एदिधिषाञ्चक्रे एदिधिषाञ्चक्राते एदिधिषाञ्चक्रिरे एदिधिषाञ्चकृषे एदिधिषाञ्चक्राथे एदिधिषाञ्चकृढ्वे एदिधिषाञ्चक्रे एदिधिषाञ्चकृवहे एधाञ्चकृमहे

luṭ एदिधिषिता एदिधिषितारौ एदिधिषितारः एदिधिषितासे

lṛṭ एदिधिषिष्यते एदिधिषिष्येते एदिधिषिष्यन्ते

loṭ एदिधिषताम् एदिधिषेताम् एदिधिषन्ताम्

laṅ ऐदिधिषत ऐदिधिषेताम् ऐदिधिषन्त

Vliṅ एदिधिषेत एदिधिषेयाताम् एदिधिषेरन् एदिधिषेथाः

Aliṅ एदिधिषिषीष्ट एदिधिषिषीयास्ताम् एदिधिषिषीरन् एदिधिषिषीष्ठाः

luṅ ऐदिधिषिष्ट ऐदिधिषिषाताम् ऐदिधिषिषत ऐदिधिषिष्ठाः ऐदिधिषिषाथाम् ऐदिधिषिढ्वम् ऐदिधिषिषि ऐदिधिषिष्वहि ऐदिधिषिष्महि

lṛṅ ऐदिधिषिष्यत ऐदिधिषिष्येताम् ऐदिधिषिष्यन्त

2 .2.5 Frequentatives:

The frequentative is formed from mono-syllabic roots of the first nine classes, beginning with any consonant. However, the roots *mūtr*, *sūc*, *sūtr* of 10th class; and the roots *aṭ, ṛ, aś* and *ūrṇu* though beginning with a vowel, take frequentative forms. The frequentatives import repetition or intensity of the action or state expressed by the root from which it is derived; as *cekrīyate* 'he does repeatedly/ intensely' from the root *kṛ*.

The freq. forms of roots implying motion does not import the usual sense but also shows tortuous motion, as *vāvrajyate* 'he walks crookedly' from the root *vraj* 'to walk'. The frequentative of the roots *lup, sad, car, jap, jabh, dah, daś, bhañj* and *gṛ* does not express the usual sense but imports censure; as *lolupyate* 'he cuts clumsily' from *lup* 'to cut'.

The frequentative base is of two kinds. Both follow the same peculiar rules of reduplication. One base is called *ātmanepada* frequentative (*yaṅanta*) as it is conjugated in the *ātmanepada* only, in which case the root takes ya before reduplication. The other base is called *parasmaipada* frequentative (*yaṅluganta*) which rejects *ya* and is conjugated in the parasmaipada only. For example: *ātmanepada* frequentative for *bhū* is *bobhūya* and *parasmaipada* frequentative base is *bobhū*.

The frequentative base is reduplicated according to the general rules of reduplication. Though there are certain special rules of reduplication and vowel changes which are equally applicable to both the *ātmanepada* and *parasmaipada* frequentative. They are:

1. If a root begins with a vowel, the following consonant is reduplicated, as *aṭ-aṭya-aṭa; aś-aśya-aśaśya*; also *arārya* and *ūrṇonūya* from *ṛ* and *ūrṇu*.
2. The vowels *i* or *u* in the reduplicative syllable takes *guṇa* substitute; as *bhū-bhūya-bhubhūya-bubhūya-bobhūya*; *nī-nīya-nīnīya-ninīya-neṇīya*; *bhid-bhidya-bhibhidya-bibhidya-bebhidya; tud-tudya-tutudya-totudya* etc.
3. The *a* in the reduplicative syllable is changed to *ā*; as *aṭāṭya* and *aśāśya* from *aṭ* and *aś*; and also *pac-pāpacya, smṛ-sāsmarya*.
4. The *a* of the reduplicative syllable of a root having a penultimate short *ṛ*, or *ṝ* which is a result of *samprasāraṇa*, is not lengthened, but the syllable *rī* is inserted between the vowel *a* of the reduplicative syllable and the first radical consonant in the *ātmanepada* frequentative base. Under the same circumstances, *lī* is inserted in the reduplicative base of the root *kḷp*. In the case of the parasmaipada freq. base, the letter *r* or the syllable *ri* or *rī* is inserted between the vowel a of the reduplicative syllable and the first radical consonant, if the root ends in short *ṛ* or has a short *ṛ* for its penultimate. *l* or *li* or *lī* is likewise inserted in the parasmai

freq. base of the root *klp*. Examples: *vṛt-vṛtya-vavṛtya-varīvṛtya; pracch-pṛcchya-paprcchya-parīpṛcchya; klp-klpya-cakīklpya; nṛt-nṛtya-nanṛtya-narīnṛtya.*

5. The *a* of the reduplicative syllable of a root ending in a nasal and the a of the roots *car, phal, jap, jabh, dah, daṁś, bhañj, paś* is not lengthened and anusvāra or the nasal of that class to which the first radical consonant belongs is inserted between the vowel a of the reduplicative syllable and the first radical consonant except in the case of roots whose final radical consonant except in the case of roots whose final radical nasal is changed or dropped before the *ya* of the freq. base. The roots *car* and *phal* after inserting a nasal, change the *a* of the following syllable to *u*.

   Examples: *car-carya-cacarya-cañcarya-cañcurya/cañcūrya*;

   phal-phalya-phaphalya-paphalya-pamphalya-pamphulya-pamphūlya;

   bhram-bhramya-bambhramya; yam-yamya-yaṁyamya.

6. The *a* of the reduplicative syllable of the roots *vañc, sraṁs, dhvaṁs, bhraṁs, kas, pat, pad* and *skand* is not lengthened and *nī* is inserted between the vowel *a* of the reduplicative syllable and the first radical consonant.

vañc	-	*vacya*	-	*vanīvacya*
sraṁs	-	*srasya*	-	*sranīsrasya*
dhvaṁs	-	*dhvasya*	-	*dhvanīdhvasya*
bhraṁs	-	*bhrasya*	-	*bhranībhrasya*
kas	-	*kasya*	-	*kanīkasya*
pat	-	*patya*	-	*panīpatya*
pad	-	*padya*	-	*panīpadya*
skand	-	*skadya*	-	*skanāskadya*

The Ātmanepada frequentative:

To form the *ātmanepada* freq. base, the syllable *ya* is added to the root; as *bhū-bhūya*; *nī-nīya*; *bhid-bhidya* etc.

Following radical vowel changes before *ya* are notable:

1. Final *i* and *u* are lengthened; as *śri-śrīya, du-dūya.*
2. Final *ṛ* preceded by a simple consonant is substituted by long *rī* and that preceded by a conjunct consonant takes *guṇa*; as *kṛ-krīya*; *smṛ-smarya.*

3. Final long *ṝ* becomes *īr*, or *ūra* if a labial or *v* precedes; as *kṛṛ-kīrya, pṛṛ-pūrya.*
4. Of roots ending in *ā*, the *ghu* verbs viz. *dā, do, de, dhe* and the roots *mā, sthā, gai, pā, hā, so, ghrā, dhmā* change the final *ā* to *ī*. The final *ā* of other roots, however, remains unchanged and the final *e, ai* and *o* except that of the roots *vye* and *hve* are changed to *ā*. The root *jyā* substitutes long *ī* , and *vye* and *hve* substitutes *ī* and *ū* for *ye* and *ve*. Ex: *dā-dīya, dhe-dhīya, gai-gīya, so-sīya*. But *cho-chāya, glai-glāya, jñā-jñāya, ghrā-ghrīya, dhmā-dhmīya, vye-vīya, hve-hūya.*
5. The roots *vyac, vyadh, syam, svap, grah, pracch, bhrasj* and *vraśc* take samprasāraṇa; as *vyac-vicya, grah-gṛhya*. The roots *yaj, vac, vap, vah, vas, vad* and *vaś* do not take samprasāraṇa. *yaj-yajya.*
6. The roots *śas* and *pyāy* become *śiṣ* and *pī*. As: *śās-śiṣya, pyāy-pīya.*
7. The roots *div, siv, ṣṭhiv, sriv* lengthen their penultimate vowel.
   div-dīvya; siv-sīvya etc.
8. A penultimate nasal is generally dropped. *śaṁs-śasya* etc.

Conjugation of the Ātmanepada freq. base:

In the active voice, the base is conjugated like the base of a root of the 4th class in Ātmanepada in the Conjugational tenses and moods. In the non-conjugational tenses and moods and in all the tenses of the passive, the base drops its ending a when the final *ya* is preceded by a vowel. But when this *ya* is preceded by a consonant, *ya* itself is dropped ; as *bobhūya* becomes *bobhūy, bobudhya* becomes *bobudh*. The base thus obtained in each case does not undergo any more changes, and the non-conjugational tenses and moods and all the tenses of the passive are formed exactly in the same way.

भू यङन्त yaNanta कर्तृवाच्य active

laṭ	बोभूयते	बोभूयेते	बोभूयन्ते	
liṭ	बोभूयाञ्चक्रे	बोभूयाञ्चक्राते	बोभूयाञ्चक्रिरे	बोभूयाञ्चकृषे
luṭ	बोभूयिता	बोभूयितारौ	बोभूयितारः	बोभूयितासे

lṛṭ	बोभूयिष्यते	बोभूयिष्येते	बोभूयिष्यन्ते			
loṭ	बोभूयताम्	बोभूयेताम्	बोभूयन्ताम्			
laṅ	अबोभूयत	अबोभूयेताम्	अबोभूयन्त			
Vliṅ	बोभूयेत	बोभूयेयाताम्	बोभूयेरन्	बोभूयेथाः		
Aliṅ	बोभूयिषीष्ट	बोभूयिषीयास्ताम्	बोभूयिषीरन्	बोभूयिषीष्ठाः		
luṅ	अबोभूयिष्ट	अबोभूयिषाताम्	अबोभूयिषत	अबोभूयिष्ठाः	अबोभूयिषाथाम्	अबोभूयिध्वम्
	अबोभूयिषि	अबोभूयिष्वहि	अबोभूयिष्महि			
lṛṅ	अबोभूयिष्यत	अबोभूयिष्येताम्	अबोभूयिष्यन्त			

Formation and conjugation of the *parasmaipada* Frequentative base:

To form the parasmaipada frequentative base, the root is reduplicated according to the general and special rules of reduplication. The base thus obtained is conjugated like the special base of a root of the 3rd class in the conjugational tenses and moods.

Augment *i* is optionally prefixed to the strong terminations beginning with a consonant, viz. before *mi, si, ti* of the present, *s* and *t* of the imperfect, and *tu* of the imperative. Before this *ī*, the penultimate short vowel does not take guṇa substitute. Examples: *bobhoti* or *bobhavīti* from *bhū*; *bebhetti* or *bebhidīti* from *bhid* etc.

In the non-conjugational tenses and moods, the base takes the *seṭ* terminations. The augment *i* is always prefixed except in the benedictive; as perfect *bobhavāñcakāra-bobhāva*; Aorist *abobhūvīt-abobhot-abobhavīt-abobhūta-abobhāvīt.*

भू यङ्लुगन्त yaNluganta कर्तृवाच्य active

laṭ	बोभवीति/बोभोति	बोभूतः	बोभुवति
	बोभवीषि/बोभोषि	बोभूथः	बोभूथ
	बोभवीमि/बोभोमि	बोभूवः	बोभूमः

liṭ बोभवाञ्चकार बोभवाञ्चक्रतुः बोभवाञ्चक्रुः बोभवाञ्चकर्थ बोभवाञ्चक्रथुः बोभवाञ्चक्र बोभवाञ्चकार बोभवाञ्चकृव बोभवाञ्चकृम

luṭ बोभविता बोभवितारौ बोभवितारः बोभवितासि बोभवितास्थः बोभवितास्थ बोभवितास्मि बोभवितास्वः बोभवितास्मः

lṛt बोभविष्यति बोभविष्यतः बोभविष्यन्ति

loṭ बोभवीतु बोभूताम् बोभुवतु बोभोहि बोभूतम् बोभूत बोभवानि बोभवाव बोभवाम बोभोतु/ बोभूतात् बोभूतात्

laṅ अबोभवीत्/ अबोभोत् अबोभूताम् अबोभवुः अबोभवीः/ अबोभोः अबोभूतम् अबोभूत अबोभवम् अबोभूव अबोभूम

Vliṅ बोभूयात् बोभूयाताम् बोभूयुः बोभूयाः बोभूयातम् बोभूयात बोभूयाम् बोभूयाव बोभूयाम

Aliṅ बोभूयात् बोभूयास्ताम् बोभूयासुः बोभूयाः बोभूयास्तम् बोभ्यास्त बोभूयासम् बोभूयास्व बोभूयास्म

luṅ अबोभूवीत् अबोभूताम् अबोभवुः अबोभूवीः अबोभूतम् अबोभूत अबोभूवम् अबोभूव अबोभूम

अबोभोत् अबोभोः

lṛṅ अबोभविष्यत् अबोभविष्यताम् अबोभविष्यन्

2.2.6 Passive forms of derived verbs:

Derived verb forms - causals, desideratives and ātmanepada frequentatives take passive forms like regular verb forms. The conjugation for the same is, for the most part, similar to that of regular verb forms. These forms with instance of *bhū* are produced below:

भू सन्नन्त sannanta कर्मवाच्य passive

laṭ बुभूष्यतें बुभूष्येते बुभूष्यन्ते

liṭ बुभूषाञ्चक्रे बुभूषाञ्चक्राते बुभूषाञ्चक्रिरे

luṭ बुभूषिता बुभूषितारौ बुभूषितारः

lṛṭ बुभूषिष्यते बुभूषिष्येते बुभूषिष्यन्ते

loṭ बुभूष्यताम् बुभूष्येताम् बुभूष्यन्ताम्

laṅ अबुभूष्यत अबुभूष्येताम् अबुभूष्यन्त

Vliṅ बुभूष्येत बुभूष्येयाताम् बुभूष्येरन्

Aliṅ बुभूषिषीष्ट बुभूषिषीयास्ताम् बुभूषिषीरन् बुभूषिषीष्ठाः बुभूषिषीयास्थाम्
बुभूषिषीध्वम् बुभूषिषीय बुभूषिषीवहि बुभूषिषीमहि

lṛṅ अबुभूषिष्यत अबुभूषिष्येताम् अबुभूषिष्यन्त

<u>भू णिजन्त nijanta कर्मवाच्य passive</u>

laṭ भाव्यते भाव्येते भाव्यन्ते

भाव्यसे भाव्येथे भाव्यध्वे

भाव्ये भाव्यावहे भाव्यामहे

liṭ भावयाञ्चक्रे भावयाञ्चक्राते भावयाञ्चक्रिरे

भावयाञ्चकृषेभावयाञ्चक्राथे भावयाञ्चकृध्वे

भावयाञ्चक्रे भावयाञ्चकृवहे भावयाञ्चकृमहे

luṭ भाविता भावितारौ भावितारः

lṛṭ भाविष्यते भाविष्येते भाविष्यन्ते

loṭ भाव्यताम् भाव्येताम् भाव्यन्ताम्

laṅ अभाव्यत अभाव्येताम् अभाव्यन्त अभाव्यथाः

Vliṅ भाव्येत भाव्येयाताम् भाव्येरन्

Aliṅ भाविषीष्ट भाविषीयास्ताम् भाविषीरन् भाविषीष्ठाः भाविषीयास्थाम् भाविषीध्वम् भाविषीय
भाविषीवहिभाविषीमहि

luṅ अभावि अभाविषाताम्, अभाविषत, अभाविष्ठाः, अभाविषाथाम्, अभाविध्वम्, अभाविषि, अभावि०, अभाविष्महि,

अभावयि० अभावयि० अभावयि० अभावयि० अभावयि० अभावयि०

अभावयिष्वहि अभावयि०

lṛṅ अभाविष्यत अभाविष्येताम् अभाविष्यन्त

<u>भू यङन्त yananta कर्मवाच्य passive</u>

laṭ बोभूय्यते बोभूय्येते बोभूय्यन्ते

liṭ बोभूयाञ्चक्रे बोभूयाञ्चक्राते बोभूयाञ्चक्रिरे बोभूयाञ्चकृषे बोभूयाञ्चक्राथे बोभूयाञ्चकृढ्वे बोभूयाञ्चक्रे बोभूयाञ्चकृवहे बोभूयाञ्चकृमहे

luṭ बोभूयिता बोभूयितारौबोभूयितारःबोभूयितासेबोभूयितासाथे बोभूयिताध्वे बोभूयिताहे बोभूयितास्वहे बोभूयितास्महे

lṛṭ बोभूयिष्यते बोभूयिष्येते बोभूयिष्यन्ते

loṭ बोभूय्यताम् बोभूय्येताम् बोभूय्यन्ताम्

laṅ अबोभूय्यतअबोभूय्येताम् अबोभूय्यन्त अबोभूय्यथाः अबोभूय्येथाम् अबोभूय्यध्वम् अबोभूय्ये अबोभूय्यावहि अबोभूय्यामहि

Vliṅ बोभूय्येत बोभूय्येयाताम् बोभूय्येरन् बोभूय्येथाः बोभूय्येयाथाम् बोभूय्येध्वम् बोभूय्येय बोभूय्येवहिबोभूय्येमहि

Aliṅ बोभूयिषीष्ट बोभूयिषीयास्ताम् बोभूयिषीरन्

luṅ अबोभूयि अबोभूयिषाताम् अबोभूयिषत अबोभूयिष्ठाः अबोभूयिषाथाम् अबोभूयिध्वम् अबोभूयिषिअबोभूयिष्वहि अबोभूयिष्महि

2.2.7 Prefixes:

Prefixes (*upasarga*-s) acquire special significance in Sanskrit morphology. Yāska, in fact, had categorized them separately in his four-fold classification. Pāṇini, however, didn't

assign them this status, though he defined and enumerated them. Prefixes (*upasarga*-s), according to Pāṇini, are enumerated in a list of words that starts with *pra* and which are termed '*upasarga*' when they are attached to verbs.[94] These *upasarga*-s are 21 or 22 in number. They have their own sense, yet their main function is to reveal the other meanings of verb roots that they are prefixed to.[95] This new meaning, as grammarians hold, is contained within the verbs but is not mentioned in the DP; just as a lamp illuminates the things of a house that already exist there, upasargas reveal different already existing meanings of the verb. This meaning sometimes is same as original, but more frequently is very different than the meaning of the combination. 22 *upasarga*-s are:

Prefix	**Principal meaning of each prefix**	**examples**
ati	Going beyond a real or imaginary limit	*atikrāmate*
adhi	Being above in place or degree	*adhirohati*
anu	After in order or manner	*anukaroti*
antar	Coming within a space or interval	*antardhīyate*
apa	Taking away in substance or kind	*apaharati*
api	Affirming of a certainty	*apigacchati*
abhi	Being present, opposite or near to	*abhisarati*
ava	Being below in place or degree	*avatarati*
ā/ ān	Bounding or limiting; also reversing	*āgacchati*
ut/ ud	Being high in place or excellent in kind	*uttiṣṭhati*
upa	Being near or next to; whence also, being less than	*upakaroti*
dur	Condition of badness, pain, difficulty, and the like	*durācarati*
ni	Being within, below or under; also, being contrary or reverse	*nipatati*
nir	Being out or exempt from; whence also affirmation, as excluding doubt	*nirvartayati*
parā	Being opposite or opposed to; whence also reverse	*parājayate*
pari	Being all round or about; whence also fullness, completeness	*paricinoti*
pra	Being before in time, place, or quality	*praviśati*
prati	Reverted, or reflected, or repeated action or condition	*prati*

[94] *prādayaḥ* 1/4/58; *upasargāḥ kriyāyoge* 1/4/59

[95] *upasargeṇa dhātvartho balādanyatra nīyate |*
prahārā''ra-saṁhāra-vihāra-parihāravat ||

vi	Being several or separate; whence also privation	*vigāyati*
sam	Being conjoined with; whence also completeness	*saṁskaroti*
su	The opposite of dur, or condition of happiness, ease and the like	*sujānāti*

Example of how prefixes cause to change the purport of verb roots; and also the pada they are conjugated in, is given below with illustrations of roots *kram* and *yam*:

kram (krām) to go, to approach, to traverse, to go up or ascend, to leap.

kram

upakram

parākram

ākrām (parasmai) – to cover, to go up or ascend

ākram (Ātmane)- 'to rise' said of a luminary

vikrām (parasmai)- to split

vikram (ātmane) – to walk or step up

prakrām (parasmai)- to go

prakram (ātmane) – to begin

upakrām (parasmai) – to come

upakram (ātmane) – to begin

yam (yacch) to check or stop

āyam (āyacch) (ātmane)- to spread, stretch when intransitive; or with 'a limb of body' or with something else than 'a literary composition' for its object.

saṁyam (saṁyacch) (ātmane)- to gather

udyam (udyacch) (ātmane)- to lift up, with something else than than 'a literary composition' for its object

udyam (udyacch) (parasmai) to try hard, to learn, with 'a literary composition' for its object.

upayam (upayacch) (ātmane) to marry

Chapter 3

Morphological Analysis of Sanskrit Verb Forms

3.1 Morphological Analysis

The aim of morphological analysis is to recognize the inner structure of the word and to break it into its morphemes in order to present a structured representation of the word. It takes the surface form as input, e.g. *bhavati,* and produces as output the analysis, i.e. *bhū* (verb root) + *ti* (*laṭ*-3rd per.-sing.). In some cases, however, a single surface form may represent more than one lexical form. For instance *bhavati* may also be analysed as *bhavat* (root) + *ṅi* (masculine-pronoun-locative-sing), or *bhavat* + *ṅi* (participle-locative-sing), or *bhavatī* + *su* (feminine-pronoun-vocative-sing). A morphological analyzer is expected to produce all the possible analyses for a given input, though to remove the ambiguity is the function of the sentence parser which takes care of the syntactico-semantic relation among the words of a sentence. Apart from morphological analysis, some other applications of natural language processing need only to retrieve the stem or root from the given word and not the complete analysis of the word. This is known as stemming which only aims at stripping off the word endings and give the stem. For some other natural language processing tasks, we need to know that two words which have different surface appearances have a similar root. For instance *bhavati, bhavataḥ, bhavanti* all are forms of common verb root *bhū* which is also known as lemma of these word-forms. To map from all these forms to a single base form is called lemmatization. Morphological parsing also needs tokenization or word segmentation which separates out different words of the given running text. This can be complicated for languages like Sanskrit which have morphophonemic operations like sandhi.

3.2 Popular Approaches

Two-level morphology is a linguistic, computationally implemented model for morphological analysis and synthesis. It was developed by Kimmo Koskenniemi, in 1980s. Before that, simple cut-and-paste methods were used only to analyze strings in particular languages, and there was no general language-independent method available. Generative phonologists of that time described morphological alternations by means of ordered rewrite rules, but it was not understood how such rules could be used for analysis.

Two-level morphology is based on three ideas:

- Rules, in this model, are symbol-to-symbol constraints that are applied in parallel, not sequentially like rewrite rules.
- The constraints can refer to the lexical context, to the surface context, or to both contexts at the same time.
- Lexical lookup and morphological analysis are performed in tandem.

Two-level model proposed parallel rules instead of successive ones, thus, avoiding the existence of intermediate stages in the derivation of single word forms. The name "two-level model" reflects the setup, where only the lexical and the surface levels "exist", and no intermediate level in between.

The two-level model is attractive as a process model, because it is based on finite state automata. It is general in the sense that the same language independent algorithm and the same computer program can operate on a wide range of languages, including highly inflected ones such as Finnish, Russian or Sanskrit. It is unrestricted in scope and it is capable of handling the whole language system as well as ordinary running text. The model is based on a lexicon that defines the word roots, inflectional morphemes and certain non-phonological alternation patterns, and on a set of parallel rules that define phonologically oriented phenomena. The rules are implemented as parallel finite state automata, and the same description can be run both in the producing and in the analyzing direction.

3.3 Strategy for present analysis

The methodology for the analysis of Sanskrit verb forms in the present work follows the analysis of Pāṇini in somewhat reverse direction. Pāṇinian analysis identifies different morphemes in any given *pada* and presents an analysis where he provides step-by-step methodology to derive a verb form from a given verb root in certain paradigms.

In the analysis of Sanskrit verb forms, we can commonly identify three morphemes involved -

- The first is the *tiṅ* termination which remains at the end of any *tiṅanta* verb form. The *tiṅ* terminations are basically 18 in number but are transformed in various paradigms as given in chapter 2. The terminations are different for every *lakāra-pada-number-person* combination and thus each one of them can uniquely identify such a combination.
- The second is the characteristic which is appended to the verb root depending on the class of the root and the lakāra it is conjugated in. For example, the conjugational classes adopt a in conjugational lakāras.
- The third one is the modified stem of the root which can be called verbal base or stem.

Though Pāṇinian analysis is far more complicated than this but to make a program for the same, one will have to include complex rules and algorithms. For this work, the analysis is reduced in the form of base + suffix, i.e. verbal base and tiṅ terminations. The basic idea is to identify every verb form in a way that it become a combination of base + suffix that is unique in itself. That means one verb form is a unique combination of base and suffix. Its base may appear in other verb form and the suffix may also come at other place, but the same combination never occurs in any other verb form. For example, bhavati = bhava+ti; bhavataḥ= bhava+taḥ; paṭhati= paṭha+ti etc. the base of the first form is appearing in the second form. Its suffix is appearing in the third one. But both of them do not appear in any other verb form. So it is a unique combination in each. If we are able to identify both of them separately, we can get the unambiguous analysis of the verb.

given verb form. The hypothesis, though cannot be claimed to be a fool-proof methodology for Sanskrit verb-form analysis, gives a satisfactory solution to the problem. In this work, however, it has been applied only in respect with *bhvādigaṇa*. So, while doing the same for rest of the classes, especially 6 non-conjugational classes, the strategy might need a lot of modifications.

3.4 The Analysis of Sanskrit Verb Forms:

This section describes the strategy to analyze Sanskrit verb forms. As stated above, Sanskrit tiṅanta forms can be analyzed into base and suffix. In a Sanskrit verb form

3.4.1 Regular Forms:

The term regular here refers to those tiṅanta verb forms which are achieved by declining primitive verb roots (non derived verb roots) in various paradigms.

3.4.1.1 Terminations for Parasmaipada

3.4.1.1.1 *laṭ lakāra(* present tense) *parasmaipada*

For *laṭ lakāra, parasmaipada* the *tiṅ* terminations are listed in the previous chapters. The terminations are almost identical to 9 basic tiṅ terminations of parasmaipada. The tables given below list these terminations in the first column. The second column contains corresponding verb endings which serve as identifiers for tiṅanta forms.

Terminations	Endings	Examples
ति	ति	भवति
तः	तः	भवतः
अन्ति	न्ति	भवन्ति
सि	सि, षि	भवसि
थः	थः	भवथः
थ	थ	भवथ

मि	◌ामि	भवामि
वः	◌ावः	भवावः
मः	◌ामः	भवामः

Table 3.1. **The terminations/endings for present tense in parasmaipada**

The termination for 3rd person plural is *anti* thus it should form verb forms like *bhavānti* etc. when affixed to base *bhava*. It is restricted and thus remains only one *a*, resulting in form *bhavanti*. For the analysis purpose, we only store *nti*, so that the remaining verbal base is similar to that of other forms. Terminations of 1st person cause lengthening of the final *a* of their verbal bases. But in our analysis, we store this *mātrā* of *ā* (◌ा) with the corresponding affixes to maintain uniformity in the verbal base throughout the nine paradigms.

3.4.1.1.2 For *loṭ lakāra* (Imperative Mood) *parasmaipada*

The terminations for the imperative mood follow almost the same pattern that of the present. Thus *bhavantu* can be explained as *bhavanti* of the present. In the 2nd person singular, the termination is deleted completely and only base remains behind. Thus we have no ending that can be identified to figure out the *lakāra*, person, and number etc. of this verb form. This is one of the limitations of present analysis discussed later.

Terminations	**Endings**	**Examples**
तु, तात्	तु, तात्	भवतु, भवतात्
ताम्	ताम्	भवताम्
अन्तु	न्तु	भवन्तु
-, तात्	-, तात्	भव, भवतात्
तम्	तम्	भवतम्
त	त	भवत

आनि	◌ानि/◌ाणि	भवानि
आव	◌ाव	भवाव
आम	◌ाम	भवाम

Table 3.2 **The terminations for Imperative mood in *parasmaipada***

The terminations of 1st person here also have *ā* attached to them but this *ā* is originally attached to them and not like the terminations of present 1st person. The singular *āni*, however, becomes *āṇi* in some cases. The singulars of 1st and 2nd persons have optional termination *tāt*. This creates another case of ambiguity.

3.4.1.1.3 For *v.liṅ lakāra* (potential mood) *parasmaipada*

The terminations for potential mood begin commonly with *ī*, which results in *e* when it is preceded by the *a*-ending base. Thus, the stored endings in the tables have *mātrā* of *e* (े) instead of *ī*. Because of Unicode Devanagari specification, the remaining base is similar to that of other conjugational tenses.

Terminations	Endings	Examples
ईत्	◌ेत्	भवेत्
ईताम्	◌ेताम्	भवेताम्
ईयुः	◌ेयुः	भवेयुः
ईस् (◌ः)	◌ेः	भवेः
ईतम्	◌ेतम्	भवेतम्
ईत	◌ेत	भवेत
ईयम्	◌ेयम्	भवेयम्
ईव	◌ेव	भवेव
ईम	◌ेम	भवेम

Table 3.3 **The terminations for potential mood in parasmaipada**

3.4.1.1.4 For *laṅ lakāra* (Imperfect tense) *parasmaipada*

The endings for imperfect tense are listed in the following table. The bases commonly adopt *a* as prefix. In 2nd person singular, visarga (ः) alone remains in the end as the identifier.

Terminations	Endings	Examples
त्	त्	अभवत्
ताम्	ताम्	अभवताम्
अन्	न्	अभवन्
स् (ः)	ः	अभवः
तम्	तम्	अभवतम्
त	त	अभवत
अम्	म्	अभवम्
व	ाव	अभवाव
म	ाम	अभवाम

Table 3.4 **The terminations for Imperfect tense in parasmaipada**

3.4.1.1.5 For *luṭ lakāra* (1st future) *parasmaipada*

Terminations in this lakāra, which are identical to endings, are simply appended to the base which may augment *i* in case the root is seṭ. The first three terminations are similar to that of ātmanepada in the same lakāra.

Terminations	Endings	Examples
ता	ता	भविता
तारौ	तारौ	भवितारौ
तारः	तारः	भवितारः
तासि	तासि	भवितासि
तास्थः	तास्थः	भवितास्थः
तास्थ	तास्थ	भवितास्थ
तास्मि	तास्मि	भवितास्मि
तास्वः	तास्वः	भवितास्वः
तास्मः	तास्मः	भवितास्मः

Table 3.5 **The terminations for 1st future in *parasmaipada***

For some roots, the terminations undergo morpho-phonemic changes. For example

सह् → सोढा, सोढारौ, सोढारः

लभ् → लब्धा, लब्धारौ, लब्धारः

क्रुश् → क्रोष्टा, क्रोष्टारौ, क्रोष्टारः

These irregular forms can be stored in the datafiles to serve as an example base for the system.

3.4.1.1.6 For *lṛṭ lakāra* (2nd future) *parasmaipada*

The terminations for first future are similar to that of present except that they adopt *sya*, which becomes *ṣya* in some cases, between the base and the terminations. Also, *seṭ* roots and some others capable of augmenting *i* in between, adopt *i*. This *i*, in present analysis, is retained with the base. The table given below lists verb forms of three roots – bhū (भू), dṛś (दृश्) and dā (दा), to illustrate the variations.

Terminations	Endings	Examples
स्यति	स्यति/ ष्यति	भविष्यति, द्रक्ष्यति , दास्यति
स्यतः	स्यतः/ ष्यतः	भविष्यतः, द्रक्ष्यतः, दास्यतः
स्यन्ति	स्यन्ति/ ष्यन्ति	भविष्यन्ति, द्रक्ष्यन्ति, दास्यन्ति
स्यसि	स्यसि/ ष्यसि	भविष्यसि, द्रक्ष्यसि, दास्यसि
स्यथः	स्यथः/ ष्यथः	भविष्यथः, द्रक्ष्यथः, दास्यथः
स्यथ	स्यथ/ ष्यथ	भविष्यथ, द्रक्ष्यथ, दास्यथ
स्यामि	स्यामि/ ष्यामि	भविष्यामि, द्रक्ष्यामि, दास्यामि
स्यावः	स्यावः/ ष्यावः	भविष्यावः, द्रक्ष्यावः, दास्यावः
स्यामः	स्यामः/ ष्यामः	भविष्यामः, द्रक्ष्यामः, दास्यामः

Table 3.6 **The terminations for Second Future in parasmaipada**

3.4.1.1.7 For *liṭ lakāra* (perfect) *parasmaipada*

Terminations	Endings	Examples
अ	-	बभूव, ममार्ज
अतुस् (ः)	तुः	बभूवतुः, ममृजतुः/ममार्जतुः
उस् (ः)	ुः	बभूवुः, ममृजुः/ममार्जुः
थ	थ/ िथ	बभूविथ, ममार्जिथ
अथुस् (ः)	थुः	बभूवथुः, ममृजथुः/ममार्जथुः
अ	-	बभूव, ममृज/ ममार्ज
अ	-	बभूव, ममार्ज
व	व/ िव	बभूविव, ममृजिव/ममृज्व/ममार्जिव
म	म/ िम	बभूविम, ममृजिम/ममृज्म/ममार्जिम

Table 3.7 **The terminations for Perfect tense in parasmaipada**

Of the nine terminations for this *lakāra*, three – *the, va* and *ma* admit the augment *i* when they are added to the roots which are capable of augmenting *i*. certain forms in this lakāra contain no tiṅ ending in the end as shown below. Only the verbal base remains behind. The table given below lists illustrations from *bhū* and *mṛj* (मृज्). Root *mṛj* shows much more variation in formation of base and over all three bases are formed.

In 2nd singular, ममार्ष्ठ is also formed which should be stored in the example base.

3.4.1.1.8 For *luṅ lakāra* (Aorist) *parasmaipada-*

This lakāra shows seven varieties of *tiṅ* endings in parasmaipada.[1] These are appended to different roots according to rules. However the first three varieties have a little difference among them. Examples are given from dā, bhū and gam. While gam is almost regular, aduḥ from dā and abhūvan, abhūvam from bhū compel us to create two bases.

Terminations	Endings	Examples
त्	त्	अदात्, अभूत् , अगमत्
ताम्	ताम्	अदाताम्, अभूताम्, अगमताम्
उः/ अन्	ुः/ न्	अदुः, अभूवन्, अगमन्
स् (ः)	ः	अदाः, अभूः, अगमः
तम्	तम्	अदातम्, अभूतम्, अगमतम्
त	त	अदात, अभूत, अगमत
अम्	म्	अदाम्, अभूवम्, अगमम्
व	व/ाव	अदाव, अभूव, अगमाव
म	म/ाम	अदाम, अभूम, अगमाम

Table 3.8 **The terminations for Aorist tense in parasmaipada (variety 1-3)**

[1] Dhaturupachandrika, Upadhye P.V., pg.231-32

Terminations	Endings	Examples
सीत्	सीत्	अवात्सीत्
स्ताम्	स्ताम्/ ताम्	अवाताम्
सुः	सुः	अवात्सुः
सीः	सीः	अवात्सीः
स्तम्	स्तम्/तम्	अवातम्
स्त	स्त/त	अवात
सम्	सम्	अवात्सम्
स्व	स्व	अवात्स्व
स्म	स्म	अवात्स्म

Table 3.9 **The terminations for Aorist tense in parasmaipada (4th variety)**

In 4th variety the ending s of roots changes to t. So we have to create alternative verb endings. The remaining varieties are simple.

Terminations	Endings	Examples
ईत्	ीत्	अव्राजीत्
इष्टाम्	िष्टाम्	अव्राजिष्टाम्
इषुः	िषुः	अव्राजिषुः
ईः	ीः	अव्राजीः
इष्टम्	िष्टम्	अव्राजिष्टम्
इष्ट	िष्ट	अव्राजिष्ट
इषम्	िषम्	अव्राजिषम्
इष्व	िष्व	अव्राजिष्व
इष्म	िष्म	अव्राजिष्म

Table3.10 **The terminations for Aorist tense in parasmaipada (5th variety)**

Terminations	Endings	Examples
सीत्	सीत्	अनंसीत्
सिष्टाम्	सिष्टाम्	अनंसिष्टाम्
सिषुः	सिषुः	अनंसिषुः
सीः	सीः	अनंसीः
सिष्टम्	सिष्टम्	अनंसिष्टम्
सिष्ट	सिष्ट	अनंसिष्ट
सिषम्	सिषम्	अनंसिषम्
सिष्व	सिष्व	अनंसिष्व
सिष्म	सिष्म	अनंसिष्म

Table 3.11 **The terminations for Aorist tense in parasmaipada (6th variety)**

Terminations	Endings	Examples
सत्	सत्/ षत्	अक्रुक्षत्
सताम्	सताम्/ षताम्	अक्रुक्षताम्
सन्	सन्/ षन्	अक्रुक्षन्
सः	सः/ षः	अक्रुक्षः
सतम्	सतम्/ षतम्	अक्रुक्षतम्
सत	सत/ षत	अक्रुक्षत
सम्	सम्/ षम्	अक्रुक्षम्
साव	साव/ षाव	अक्रुक्षाव
साम	साम/ षाम	अक्रुक्षाम

Table 3.12 **The terminations for Aorist tense in parasmaipada (7th variety)**

3.4.1.1.9 For *āśīrliṅ lakāra* (benedictive) *parasmaipada-*

The terminations for this mood are simply concatenative in nature.

Terminations	Endings	Examples
यात्	यात्	भूयात्
यास्ताम्	यास्ताम्	भूयास्ताम्
यासुः	यासुः	भूयासुः
याः	याः	भूयाः
यास्तम्	यास्तम्	भूयास्तम्
यास्त	यास्त	भूयास्त
यासम्	यासम्	भूयासम्
यास्व	यास्व	भूयास्व
यास्म	यास्म	भूयास्म

Table 3.13 **The terminations for benedictive mood in parasmaipada**

3.4.1.1.10 For *lṛṅ lakāra* (conditional mood) *parasmaipada-*

Terminations	Endings	Examples
स्यत्	स्यत् / ष्यत्	अभविष्यत्, अदास्यत् , अद्रक्ष्यत्
स्यताम्	स्यताम् / ष्यताम्	अभविष्यताम् , अदास्यताम्, अद्रक्ष्यताम्
स्यन्	स्यन् / ष्यन्	अभविष्यन् , अदास्यन् , अद्रक्ष्यन्
स्यः	स्यः / ष्यः	अभविष्यः , अदास्यः, अद्रक्ष्यः
स्यतम्	स्यतम् / ष्यतम्	अभविष्यतम्, अदास्यतम्, अद्रक्ष्यतम्
स्यत	स्यत / ष्यत	अभविष्यत , अदास्यत, अद्रक्ष्यत

स्यम्	स्यम् / ष्यम्	अभविष्यम् , अदास्यम्, अद्रक्ष्यम्
स्याव	स्याव / ष्याव	अभविष्याव , अदास्याव, अद्रक्ष्याव
स्याम	स्याम / ष्याम	अभविष्याम, अदास्याम, अद्रक्ष्याम

Table 3.14 **The terminations for Conditional mood in parasmaipada**

3.4.1.2 Terminations for Atmanepada:

The terminations of ātmanepada are given below. They are also affixed to the passive forms as well without much variation.

3.4.1.2.1 For *laṭ lakāra ātmanepada-*

The terminations for laṭ lakāra are given in the table. The 3rd and 2nd person dual forms and the 1st person singular begin with *i*, resulting in *e* when added to an *a*-ending base. The *mātrā* of *e* (े) is stored with the ending. Terminations for 1st person dual and plural add *ā* before them.

Terminations	Endings	Examples
ते	ते	एधते
इते	ेते	एधेते
अन्ते	न्ते	एधन्ते
से	से	एधसे
इथे	ेथे	एधेथे
ध्वे	ध्वे	एधध्वे
इ	े	एधे
वहे	ावहे	एधावहे
महे	ामहे	एधामहे

Table 3.15 **The terminations for Present tense in ātmanepada**

3.4.1.2.2 For *loṭ lakāra ātmanepada*

Terminations	Endings	Examples
ताम्	ताम्	एधताम्
इताम्	ेताम्	एधेताम्
अन्ताम्	न्ताम्	एधन्ताम्
स्व	स्व	एधस्व
इथाम्	ेथाम्	एधेथाम्
ध्वम्	ध्वम्	एधध्वम्
ऐ	ै	एधै
आवहै	ावहै	एधावहै
आमहै	ामहै	एधामहै

Table 3.16 **The terminations for Imperative mood in ātmanepada**

3.4.1.2.3 For *v.liṅ lakāra ātmanepada*

The terminations for potential mood commonly begin with *ī*. The result is, as in the other places, *e*, mātrā of which is stored with the endings.

Terminations	Ending	Examples
ईत	ेत	एधेत
ईयाताम्	ेयाताम्	एधेयाताम्
ईरन्	ेरन्	एधेरन्
ईथाः	ेथाः	एधेथाः
ईयाथाम्	ेयाथाम्	एधेयाथाम्

ईध्वम्	ेध्वम्	एधेध्वम्
ईय	ेय	एधेय
ईवहि	ेवहि	एधेवहि
ईमहि	ेमहि	एधेमहि

Table 3.17 **The terminations for potential mood in ātmanepada**

3.4.1.2.4 For *laṅ lakāra ātmanepada:*

The terminations for laṅ lakāra in Atmanepada are listed below:

Terminations	**Endings**	**Examples**
त	त	ऐधत
इताम्	ेताम्	ऐधेताम्
अन्त	न्त	ऐधन्त
थास् (ः)	थाः	ऐधथाः
इथाम्	ेथाम्	ऐधेथाम्
ध्वम्	ध्वम्	ऐधध्वम्
इ	े	ऐधे
वहि	ावहि	ऐधावहि
महि	ामहि	ऐधामहि

Table 3.18 **The terminations for Imperfect Tense in ātmanepada**

3.4.1.2.5 For *luṭ lakāra ātmanepada*

Terminations	Endings	Examples
ता	ता	एधिता
तारौ	तारौ	एधितारौ
तारः	तारः	एधितारः
तासे	तासे	एधितासे
तासाथे	तासाथे	एधितासाथे
ताध्वे	ताध्वे	एधिताध्वे
ताहे	ताहे	एधिताहे
तास्वहे	तास्वहे	एधितास्वहे
तास्महे	तास्महे	एधितास्महे

Table 3.19 **The terminations for the First Future in ātmanepada**

3.4.1.2.6 For *lṛṭ lakāra ātmanepada*

Terminations	Endings	Examples
स्यते	स्यते/ ष्यते	एधिष्यते
स्येते	स्येते/ ष्येते	एधिष्येते
स्यन्ते	स्यन्ते/ ष्यन्ते	एधिष्यन्ते
स्यसे	स्यसे/ ष्यसे	एधिष्यसे
स्येथे	स्येथे/ ष्येथे	एधिष्येथे
स्यध्वे	स्यध्वे/ ष्यध्वे	एधिष्यध्वे
स्ये	स्ये/ ष्ये	एधिष्ये
स्यावहे	स्यावहे/ ष्यावहे	एधिष्यावहे
स्यामहे	स्यामहे/ ष्यामहे	एधिष्यामहे

Table 3.20 **The terminations for the second future tense in ātmanepada**

3.4.1.2.7 For *liṭ lakāra ātmanepada* –

Perfect tense has two varieties: Reduplicative and Periphrastic.

The terminations of the first one are listed below. The illustrations are given from root kṣam.

Terminations	Endings	Examples
ए	े	चक्षमे
आते	ाते	चक्षमाते
इरे	िरे	चक्षमिरे
से	से, षे, िषे	चक्षमिषे/ चक्षंसे
आथे	ाथे	चक्षमाथे
ध्वे	ध्वे/ढ्वे, िध्वे/िढ्वे	चक्षमिध्वे/ चक्षन्ध्वे
ए	े	चक्षमे
वहे	वहे, िवहे	चक्षमिवहे/ चक्षण्वहे
महे	महे, िमहे	चक्षमिमहे/ चक्षण्महे

Table 3.21 **The terminations for the reduplicative perfect tense in ātmanepada**

Periphrastic perfect:

Periphrastic perfect is formed in three ways. The perfect forms of *as*, *kṛ* and *bhū* are appended to verb roots with an insertion of *ām* before terminations.

Terminations	Endings	Examples
आस	ामास	एधामास
आसतुः	ामासतुः	एधामासतुः
आसुः	ामासुः	एधामासुः
आसिथ	ामासिथ	एधामासिथ

आसथुः	◌ामासथुः	एधामासथुः
आस	◌ामास	एधामास
आस	◌ामास	एधामास
आसिव	◌ामासिव	एधामासिव
आसिम	◌ामासिम	एधामासिम

Table 3.22 **The terminations for Periphrastic perfect in *ātmanepada* (1st variety)**

Terminations	Endings	Examples
चक्रे	◌ाञ्चक्रे	एधाञ्चक्रे
चक्राते	◌ाञ्चक्राते	एधाञ्चक्राते
चक्रिरे	◌ाञ्चक्रिरे	एधाञ्चक्रिरे
चकृषे	◌ाञ्चकृषे	एधाञ्चकृषे
चक्राथे	◌ाञ्चक्राथे	एधाञ्चक्राथे
चकृढ्वे	◌ाञ्चकृढ्वे	एधाञ्चकृढ्वे
चक्रे	◌ाञ्चक्रे	एधाञ्चक्रे
चकृवहे	◌ाञ्चकृवहे	एधाञ्चकृवहे
चकृमहे	◌ाञ्चकृमहे	एधाञ्चकृमहे

Table 3.23 **The terminations for Periphrastic perfect in *ātmanepada* (2st variety)**

Terminations	Endings	Examples
बभूव	◌ाम्बभूव	एधाम्बभूव
बभूवतुः	◌ाम्बभूवतुः	एधाम्बभूवतुः
बभूवुः	◌ाम्बभूवुः	एधाम्बभूवुः
बभूविथ	◌ाम्बभूविथ	एधाम्बभूविथ

बभूवथुः	◌ाम्बभूवथुः	एधाम्बभूवथुः
बभूव	◌ाम्बभूव	एधाम्बभूव
बभूव	◌ाम्बभूव	एधाम्बभूव
बभूविव	◌ाम्बभूविव	एधाम्बभूविव
बभूविम	◌ाम्बभूविम	एधाम्बभूविम

Table 3.24 **The terminations for Periphrastic perfect in *ātmanepada* (3st variety)**

3.4.1.2.8 For *luṅ lakāra ātmanepada* –

The Aorist or luṅ lakāra shows multiple varieties. In ātmanepada, there are four varieties.

Terminations	Endings	Examples
इष्ट	◌िष्ट	ऐधिष्ट
इषाताम्	◌िषाताम्	ऐधिषाताम्
इषत	◌िषत	ऐधिषत
इष्ठाः	◌िष्ठाः	ऐधिष्ठाः
इषाथाम्	◌िषाथाम्	ऐधिषाथाम्
इढ्वम्, इध्वम्	◌िढ्वम्, ◌िध्वम्	ऐधिढ्वम्
इषि	◌िषि	ऐधिषि
इष्वहि	◌िष्वहि	ऐधिष्वहि
इष्महि	◌िष्महि	ऐधिष्महि

Table 3.25 **The terminations for Aorist in ātmanepada**

Terminations	Endings	Examples
त	त	अह्वत / अशिश्रियत
इताम्	ेताम्	अह्वेताम् / अशिश्रियेताम्
अन्त	न्त	अह्वन्त / अशिश्रियन्त
थास्	थाः	अह्वथाः / अशिश्रियथाः
इथाम्	ेथाम्	अह्वेथाम् / अशिश्रियेथाम्
ध्वम्	ध्वम्, ढ्वम्	अह्वध्वम् / अशिश्रियध्वम्
इ	े	अह्वे / अशिश्रिये
वहि	ावहि	अह्वावहि / अशिश्रियावहि
महि	ामहि	अह्वामहि / अशिश्रियामहि

Table 3.26 **The terminations for Aorist in ātmanepada**

Terminations	Endings	Examples
स्त	स्त	अत्रास्त
साताम्	साताम्	अत्रासाताम्
सत	सत	अत्रासत
स्थाः	स्थाः	अत्रास्थाः
साथाम्	साथाम्	अत्रासाथाम्
ध्वम्	ध्वम्	आत्राध्वम्
सि	सि	अत्रासि
स्वहि	स्वहि	अत्रास्वहि
स्महि	स्महि	अत्रास्महि

Table 3.27 **The terminations for Aorist in ātmanepada**

Terminations	Endings	Examples
सत	सत/ षत	अलिक्षत
साताम्	साताम्/ षाताम्	अलिक्षाताम्
सन्त	सन्त/ षन्त	अलिक्षन्त
सथाः	सथाः/ षथाः	अलिक्षथाः
साथाम्	साथाम्/ षाथाम्	अलिक्षाथाम्
सध्वम्	सध्वम्/ षध्वम्	अलिक्षध्वम्
सि	सि/ षि	अलिक्षि
सावहि	सावहि/ षावहि	अलिक्षावहि
सामहि	सामहि/ षामहि	अलिक्षामहि

Table 3.28 **The terminations for Aorist in ātmanepada**

3.4.1.2.9 For *āśīrliṅ lakāra Atmanepada* –

The terminations for benedictive mood in *ātmanepada* begin with consonant 's'. These either remain the same in the usage and are affixed to the base as such; or the initial s is altered to ṣ in respective environment. Another change is that the *dh* of *sīdhvam* is changed to Dh PS: ?? by the same rule.

Terminations	Endings	Examples
सीष्ट	सीष्ट / षीष्ट	एधिषीष्ट
सीयास्ताम्	सीयास्ताम्/ षीयास्ताम्	एधिषीयास्ताम्
सीरन्	सीरन् /षीरन्	एधिषीरन्
सीष्ठाः	सीष्ठाः/ षीष्ठाः	एधिषीष्ठाः
सीयास्थाम्	सीयास्थाम्/ षीयास्थाम्	एधिषीयास्थाम्
सीध्वम्	सीध्वम्/ षीध्वम्/ /	एधिषीध्वम्

सीय	सीय/ षीय	एधिषीय
सीवहि	सीवहि/ षीवहि	एधिषीवहि
सीमहि	सीमहि/ षीमहि	एधिषीमहि

Table 3.29 **The terminations for benedictive mood in *ātmanepada***

3.4.1.2.10 For ***lṛṅ lakāra Atmanepada*** –

The ātmanepada terminations in *lṛṅ lakāra* are listed below.

Terminations	**Endings**	**Examples**
स्यत	स्यत/ ष्यत	ऐधिष्यत
स्येताम्	स्येताम्/ ष्येताम्	ऐधिष्येताम्
स्यन्त	स्यन्त / ष्यन्त	ऐधिष्यन्त
स्यथाः	स्यथाः /ष्यथाः	ऐधिष्यथाः
स्येथाम्	स्येथाम्/ ष्येथाम्	ऐधिष्येथाम्
स्यध्वम्	स्यध्वम्/ ष्यध्वम्	ऐधिष्यध्वम्
स्ये	स्ये/ ष्ये	ऐधिष्ये
स्यावहि	स्यावहि/ ष्यावहि	ऐधिष्यावहि
स्यामहि	स्यामहि/ ष्यामहि	ऐधिष्यामहि

Table 3.30 **The terminations for *lṛṅ lakāra Atmanepada***

3.4.2 Passive and Impersonal forms:

The passives, as mentioned earlier, are rather simpler derivatives of primitive roots. Whatever be the class of the root, the passives are formed by a common formula. The root is augmented with *ya* and *ātmanepada* terminations are added to it.

Terminations	Endings	Examples
ते	ते	भूयते, एध्यते
त	त	अभूयत, ाइधयत
ताम्	ताम्	भूयताम्, स्पद्ध्यताम्
एत	ेत	भूयेत, स्पद्ध्येत
आमास	ामास	बोभूयामास, स्पर्द्धामास
इष्ट	िष्ट	अबोभूयिष्ट, अस्पर्द्धिष्ट
ष्यते	ष्यते	भूयिष्यते, स्पर्द्धिष्यते
ता	ता	भूयिता, स्पर्द्धिता
ष्यत	ष्यत	अभूयिष्यत, अस्पर्द्धिष्यत
इषीष्ट	िषीष्ट	बोभूयिषीष्ट, स्पर्द्धिषीष्ट

Table 3.31 **The terminations for *lṛṅ lakāra Atmanepada***

3.4.3 For Derived verbs:

3.4.3.1 Causals:

Causals of verb roots are conjugated like the verbs of the tenth class which is quite similar to the conjugation of the first class. So, the causals adopt the same terminations, in most cases, as are adopted by the regular verb roots. This can be illustrated by the following table. However the Aorist deviates from the trend as is shown later.

Terminations	Endings	Examples
ति	ति	भावयति
अत्	त्	अभावयत्
तु	तु	भावयतु
ईत्	ेत्	भावयेत्
चकार	ाञ्चकार	भावयाञ्चकार
स्यति	स्यति/ष्यति	भावयिष्यति
ता	ता	भावयिता
ष्यत्	ष्यत्	अभावयिष्यत्
यात्	यात्	भाव्यात्

Table 3.32 **The terminations for *lṛṅ lakāra Atmanepada***

The Aorist of Causals are different from regular verb forms. The endings are rather similar to the simple past (laṅ) of the regular as shown below:

Terminations	Endings	Examples
त्	त्	अबीभवत्
ताम्	ताम्	अबीभवताम्
न्	न्	अबीभवन्
:	:	अबीभवः
तम्	तम्	अबीभवतम्
त	त	अबीभवत
म्	म्	अबीभवम्
आव	ाव	अबीभवाव
आम	ाम	अबीभवाम

Table 3.33 **The terminations for *Aorist* of the causals**

3.4.3.2 Desideratives:

Desideratives Parasmaipada

Terminations	Endings	Examples
ति	ति	बुभूषति
तः	तः	बुभूषतः
अन्ति	न्ति	बुभूषन्ति
सि	सि, षि	बुभूषसि
थः	थः	बुभूषथः
थ	थ	बुभूषथ
मि	ामि	बुभूषामि
वः	ावः	बुभूषावः
मः	ामः	बुभूषावः

Table 3.34 **The terminations for Desideratives Parasmaipada**

Desiderative ātmanepada

Terminations	Endings	Examples
ते	ते	बुभूषते
त	त	अबुभूषत
ताम्	ताम्	अबुभूषताम्
एत	ेत	अबुभूषेत
ष्यते	ष्यते	बुभूषिष्यते, स्पर्द्धिष्यते
ता	ता	बुभूषिता, स्पर्द्धिता

Table 3.35 **The terminations for Desiderative ātmanepada**

3.4.3.3 Frequentatives:

The terminations for frquentatives are listed below:

Terminations	Endings	Examples
ते	ते	बोभूयते, पास्पद्ध्यते
त	त	अबोभूयत, अपास्पद्ध्यत
ताम्	ताम्	बोभूयताम्, पास्पद्ध्यताम्
एत	ेत	बोभूयेत, पास्पद्ध्येत
आमास	ामास	बोभूयामास, पास्पद्धामास
इष्ट	िष्ट	अबोभूयिष्ट, अपास्पद्धिष्ट
ष्यते	ष्यते	बोभूयिष्यते, पास्पद्धिष्यते
ता	ता	बोभूयिता, पास्पद्धिता
ष्यत	ष्यत	अबोभूयिष्यत, अपास्पद्धिष्यत
इषीष्ट	िषीष्ट	बोभूयिषीष्ट, पास्पद्धिषीष्ट

Table 3.36 **The terminations for *Frequentative Atmanepada***

3.4.4 The verbal bases:

The formation of the verbal bases has been explained in the previous chapter. The verbal base is the augmented form of the verb-root (*dhātu*), the meaning bearing entity. Thus it contains the core meaning of the verb in itself. The base is the modified string, to which the *tiṅ* terminations are affixed to form the final *tiṅanta*. Base is almost identical to *aṅga* of Sanskrit grammar. Sanskrit grammatical analysis defines *aṅga* as the entire string (*śabda-svarūpa*) to which a suffix (*pratyaya*) is attached. For example, *tiṅ* termination *ti* (*tip*) is affixed to root *bhū* in *laṭ*, 1st person, singular. The characteristic of *bhvādigaṇa* - *śap* (*a*) is inserted between the root and the termination. The root undergoes morphophonemic changes and is modified to become *bhava* (*bhav+a*). Now, this entire string consisting up of (verb root) + (characteristic) is termed *aṅga,* to which affix *ti* is appended.

The base involves two or three main things. The first is the change which occurs due to the environment, i.e. the *guṇa* or *vṛddhi*. The second is the characteristic which is a result of the class to which root belongs to.

Bases for the conjugational tenses:

The ten classes of Sanskrit primitive verb roots are classified in two categories on the basis of the characteristics that they adopt in the four of *lakāras*, i.e. laṭ, loṭ, laṅ and *vidhiliṅ*. Among these classes, four classes- 1st, 4th, 6th and 10th show similarity and all the verbal bases constructed out of the verb roots of these classes end in a. So, for bhvādigaṇa also, we have most of the verbal bases ending in a. This feature simplifies the technique that is to be used in this work because it nullifies the effect of the morphophonemic changes that occur in the root due to certain operations. We have variations in the formation of different bases from the verb root, yet in the end, we obtain all of them harmonized with *a* ending.

A table on next page illustrates the storing of bases of *bhū* in various *lakāras*.

lakāra	**bases**	**Examples**
लट्	भव	भवति
लोट्	भव	भवतु
विधिलिङ्	भव	भवेत्
लङ्	अभव	अभवत्
लिट्	बभूव	बभूव, बभूवतुः
आशीर्लिङ्	भू	भूयात्
लृट्	भवि	भविष्यति
लुट्	भवि	भविता
लुङ्	अभू,अभूव	अभूत्, अभूवन्
लृङ्	अभवि	अभविष्यत्

Table 3.38 **The bases of bhū in different lakāras**

3.5 Verb analysis examples

The morphological analysis for regular verb-forms of Sanskrit language given above can be illustrated with example of bhū and edha. The verb forms of bhū have been listed in the previous chapter. Here we are analyzing that data according to the analysis methodology presented by us.

Bhū active

laṭ lakāra parasmaipada

भवति = भव + ति	भवतः = भव + तः	भवन्ति = भव + न्ति
भवसि = भव + सि	भवथः = भव + थः	भवथ = भव + थ
भवामि = भव + ◌ामि	भवावः = भव + ◌ावः	भवामः = भव + ◌ामः

lit

बभूव= बभूव + -	बभूवतुः = बभूव + तुः	बभूवुः = बभूव + ◌ुः
बभूविथ= बभूव + ◌िथ	बभूवथुः= बभूव + थुः	बभूव = बभूव + -
बभूव= बभूव + -	बभूविव= बभूव + ◌िव	बभूविम = बभूव + ◌िम

lut

भविता = भवि +ता	भवितारौ = भवि + तारौ	भवितारः = भवि + तारः
भवितासि = भवि + तासि	भवितास्थः = भवि + तास्थः	भवितास्थ = भवि + तास्थ
भवितास्मि = भवि + तास्मि	भवितास्वः = भवि + तास्वः	भवितास्मः = भवि + तास्मः

lrt

भविष्यति = भवि + ष्यति	भविष्यतः = भवि + ष्यतः	भविष्यन्ति = भवि + ष्यन्ति
भविष्यसि = भवि + ष्यसि	भविष्यथः = भवि + ष्यथः	भविष्यथ = भवि + ष्यथ
भविष्यामि = भवि + ष्यामि	भविष्यावः = भवि + ष्यावः	भविष्यामः = भवि + ष्यामः

lot

भवतु/ भवतात् = भव + तु/ तात्	भवताम् = भव + ताम्	भवन्तु = भव + न्तु

भव/ भवतात् = भव +- /तात् भवतम् = भव + तम् भवत = भव + त

भवानि = भव +ानि भवाव = भव +ाव भवाम = भव +ाम

laṅ

अभवत् = अभव + त् अभवताम् = अभव +ताम् अभवन् = अभव + न्

अभवः = अभव + ः अभवतम् = अभव +तम् अभवत = अभव + त

अभवम् = अभव + म् अभवाव = अभव + ाव अभवाम = अभव +ाम

Vliṅ

भवेत् = भव +ेत् भवेताम् = भव + ेताम् भवेयुः = भव +ेयुः

भवेः = भव +ेः भवेतम् = भव + ेतम् भवेत = भव +ेत

भवेयम् = भव +ेयम् भवेव = भव + ेव भवेम = भव +ेम

Aliṅ

भूयात् = भू + यात् भूयास्ताम् = भू + यास्ताम् भूयासुः = भू + यासुः

भूयाः = भू + याः भूयारस्तम् = भू + यास्तम् भूयास्त = भू + यास्त

भूयासम् = भू + यासम् भूयास्व = भू + यास्व भूयास्म = भू + यास्म

luṅ

अभूत् = अभू + त् अभूताम् = अभू + ताम् अभूवन् = अभू +वन्

अभूः = अभू + ः अभूतम् = अभू + तम् अभूत = अभू +त

अभूवम् = अभू +वम् अभूव = अभू + व अभूम = अभू +म

lriṅ

अभविष्यत् = अभवि + ष्यत् अभविष्यताम् = अभवि ष्यताम् अभविष्यन्= अभवि + ष्यन्

अभविष्यः= अभवि + ष्यः अभविष्यतम् = अभवि +ष्यतम् अभविष्यत = अभवि + ष्यत

अभविष्यम् = अभवि + ष्यम् अभविष्याव = अभवि +ष्याव अभविष्याम= अभवि + ष्याम

Bhū passive

lat

भूयते = भूय +ते भूयेते= भूय +ेते भूयन्ते= भूय +न्ते

भूयसे= भूय +से भूयेथे= भूय +ेथे भूयध्वे= भूय +ध्वे

भूये= भूय +े भूयावहे= भूय +ावहे भूयामहे= भूय +ामहे

lit

बभूवे=बभूव + े बभूवाते=बभूव +ाते बभूविरे=बभूव +िरे

बभूविषे=बभूव + िषे बभूवाथे=बभूव + ाथे बभूविढ्वे,ध्वे=बभूव+िध्वे/ िढ्वे

बभूवे=बभूव +े बभूविवहे=बभूव +िवहे बभूविमहे=बभूव +िमहे

lut

भविता=भवि+ता भवितारौ=भवि+तारौ भवितारः=भवि+तारः

lrt

भविष्यते = भवि + ष्यते भविष्येते = भवि + ष्येते भविष्यन्ते = भवि + ष्यन्ते

भविष्यसे = भवि + ष्यसे भविष्येथे = भवि + ष्येथे भविष्यध्वे = भवि + ष्यध्वे

भविष्ये = भवि + ष्ये भविष्यावहे = भवि + ष्यावहे भविष्यामहे = भवि +ष्यामहे

lot

भूयताम्= भूय +ताम् भूयेताम्= भूय +ेताम् भूयन्ताम्= भूय +न्ताम्

भूयस्व= भूय +स्व भूयेथाम्= भूय +ेथाम् भूयध्वम्= भूय +ध्वम्

भूयै= भूय + ै भूयावहै= भूय +ावहै भूयामहै= भूय +ामहै

laṅ

अभूयत = अभूय +त अभूयेताम्= अभूय + ेताम् अभूयन्त= अभूय +न्त

अभूयथाः = अभूय + थाः अभूयेथाम्= अभूय +ेथाम् अभूयध्वम् = अभूय +ध्वम्

अभूये = अभूय +े अभूयावहि = अभूय +ावहि अभूयामहि = अभूय +ामहि

Vliṅ

भूयेत= भूय +ेत भूयेयाताम्= भूय +ेयाताम् भूयेरन्= भूय +ेरन्

भूयेथाः= भूय +ेथाः भूयेयाथाम्= भूय +ेयाथाम् भूयेध्वम्= भूय +ेध्वम्

भूयेय= भूय +ेय भूयेवहि= भूय +ेवहि भूयेमहि= भूय +ेमहि

Aliṅ

भविषीष्ट = भवि+षीष्ट भविषीयास्ताम् भविषीरन्

भविषीष्ठाः भविषीयास्थाम् भविषीध्वम्

भविषीय भविषीवहि भविषीमहि

lriN

अभविष्यत = अभवि +ष्यत अभविष्येताम् = अभवि +ष्येताम् अभविष्यन्त = अभवि +ष्यन्त

अभविष्यथाः = अभवि +ष्यथाः अभविष्येथाम् = अभवि +ष्येथाम् अभविष्यध्वम् = अभवि +ष्यध्वम्

अभविष्ये = अभवि + ष्ये अभविष्यावहि = अभवि +ष्यावहि अभविष्यामहि = अभवि +ष्यामहि

भू णिजन्त Nijanta कर्तृवाच्य active परस्मैपद

lat

भावयति = भावय + ति भावयतः = भावय + तः भावयन्ति = भावय + न्ति

lit

भावयाञ्चकार=भावय+ाञ्चकार भावयाञ्चक्रतुः=भावय+ाञ्चक्रतुः भावयाञ्चक्रुः=भावय+ाञ्चक्रुः

भावयाञ्चकर्थ=भावय+ाञ्चकर्थ भावयाञ्चक्रथुः=भावय+ाञ्चक्रथुः भावयाञ्चक्र=भावय+ाञ्चक्र

भावयाञ्चकार = भावय+ाञ्चकार भावयाञ्चकृव = भावय +ाञ्चकृव भावयाञ्चकृम=भावय+ाञ्चकृम

lut

भावयिता = भावयि + ता भावयितारौ = भावयि +तारौ भावयितारः = भावयि + तारः

lrt

भावयिष्यति = भावयि +ष्यति भावयिष्यतः = भावयि +ष्यतः भावयिष्यन्ति = भावयि +ष्यन्ति

lot

भावयतु,तात् = भावय + तु/तात् भावयताम् = भावय +ताम् भावयन्तु= भावय +न्तु

laN

अभावयत्= अभावय +त् अभावयताम्= अभावय +ताम् अभावयन्= अभावय +न्

VliN

भावयेत् = भावय +ेत् भावयेताम्= भावय +ेताम् भावयेयुः = भावय +ेयुः

AliN

भाव्यात् = भाव्+यात् भाव्यास्ताम्=भाव्+यास्ताम् भाव्यासुः=भाव्+यासुः

luN

अबीभवत् = अबीभव +त् अबीभवताम् = अबीभव +ताम् अबीभवन् = अबीभव +न्

अबीभवः = अबीभव +ः अबीभवतम् = अबीभव +तम् अबीभवत = अबीभव +त

अबीभवम् = अबीभव +म् अबीभवाव = अबीभव +ाव अबीभवाम = अबीभव+ाम

lṛṅ

अभावयिष्यत्=अभावयि+ष्यत् अभावयिष्यताम्=अभावयि+ष्यताम् अभावयिष्यन्=अभावयि+ष्यन्

<u>भू णिजन्त Nijanta कर्तृवाच्य active आत्मनेपद</u>

lat

भावयते= भावय +ते भावयेते= भावय +ेते भावयन्ते= भावय +न्ते

lit

भावयाञ्चक्रे = भावय + ाञ्चक्रे भावयाञ्चक्राते= भावय +ाञ्चक्राते भावयाञ्चक्रिरे = भावय +ाञ्चक्रिरे

lut

भावयिता= भावयि +ता भावयितारौ= भावयि +तारौ भावयितारः= भावयि +तारः

lrt

भावयिष्यते= भावयि +ष्यते भावयिष्येते= भावयि +ष्येते भावयिष्यन्ते= भावयि +ष्यन्ते

lot

भावयताम् = भावय + ताम् भावयेताम् = भावय + ेताम् भावयन्ताम् = भावय +न्ताम्

laN

अभावयत= अभावय + त अभावयेताम्= अभावय + ेताम् अभावयन्त = अभावय + न्त

VliN

भावयेत= भावय +ेत भावयेयाताम् = भावय +ेयाताम् भावयेरन् = भावय +ेरन्

AliN

भावयिषीष्ट = भावयि +षीष्ट भावयिषीयास्ताम् = भावयि +षीयास्ताम् भावयिषीरन् = भावयि + षीरन्

luN

अबीभवत्=अबीभव + त् अबीभवेताम् =अबीभव + ेताम् अबीभवन्त =अबीभव +न्त

अबीभवथाः=अबीभव + ः अबीभवेथाम्=अबीभव + ेथाम् अबीभवध्वम् =अबीभव +ध्वम्

अबीभवे=अबीभव + े अबीभवावहि=अबीभव +ावहि अबीभवामहि=अबीभव +ामहि

lriN

अभावयिष्यत = अभावयि + ष्यत अभावयिष्येताम्= अभावयि + ष्येताम् अभावयिष्यन्त= अभावयि + ष्यन्त

भू सन्नन्त sannanta कर्तृवाच्य active परस्मैपद

lat

बुभूषति = बुभूष + ति बुभूषतः = बुभूष + तः बुभूषन्ति= बुभूष + न्ति

lit

बुभूषाञ्चकार = बुभूष+ाञ्चकार बुभूषाञ्चक्रतुः =बुभूष+ाञ्चक्रतुः बुभूषाञ्चक्रुः=बुभूष+ ाञ्चक्रुः

lut

बुभूषिता=बुभूषि+ता बुभूषितारौ =बुभूषि+तारौ बुभूषितारः=बुभूषि+तारः

lrt

बुभूषिष्यति=बुभूषि+ष्यति बुभूषिष्यतः =बुभूषि+ष्यतः बुभूषिष्यन्ति =बुभूषि+ष्यन्ति

lot

बुभूषतु,तात् = बुभूष+तु,तात् बुभूषताम् =बुभूष+ताम् बुभूषन्तु =बुभूष+न्तु

laN

अबुभूषत्=अबुभूष+त् अबुभूषताम्=अबुभूष+ताम् अबुभूषन् =अबुभूष+न्

VliN

बुभूषेत् =बुभूष+ेत् बुभूषेताम् =बुभूष+ेताम् बुभूषेयुः =बुभूष+ेयुः

AliN

बुभूष्यात् = बुभूष् + यात् बुभूष्यास्ताम् = बुभूष् + यास्ताम् बुभूष्यासुः = बुभूष् +यासुः

luN

अबुभूषीत् = अबुभूष+ीत् अबुभूषिष्टाम् = अबुभूष+िष्टाम् अबुभूषिषुः = अबुभूष+िषुः

lriN

अबुभूषिष्यत् =अबुभूषि+ष्यत् अबुभूषिष्यताम्=अबुभूषि+ ष्यताम् अबुभूषिष्यन् =अबुभूषि+ष्यन्

<u>भू सन्नन्त sannanta कर्तृवाच्य active आत्मनेपद</u>

lat

बुभूषते=बुभूष+ते बुभूषेते=बुभूष+ेते बुभूषन्ते =बुभूष+न्ते

lit

बुभूषाञ्चक्रे=बुभूष+ाञ्चक्रे बुभूषाञ्चक्राते=बुभूष+ ाञ्चक्राते बुभूषाञ्चक्रिरे=बुभूष+ ाञ्चक्रिरे

lut

बुभूषिता=बुभूषि+ ता बुभूषितारौ=बुभूषि+तारौ बुभूषितारः=बुभूषि+तारः

बुभूषितासे=बुभूषि+ तासे बुभूषितासाथे=बुभूषि+तासाथे बुभूषिताध्वे=बुभूषि+ताध्वे

lrt

बुभूषिष्यते=बुभूषि+ष्यते बुभूषिष्येते=बुभूषि+ष्येते बुभूषिष्यन्ते=बुभूषि+ष्यन्ते

lot

बुभूषताम्=बुभूष+ताम् बुभूषेताम्=बुभूष+ेताम् बुभूषन्ताम्=बुभूष+न्ताम्

laN

अबुभूषत=अबुभूष+ त अबुभूषेताम्=अबुभूष+ ेताम् अबुभूषन्त=अबुभूष+ न्त

VliN

बुभूषेत=बुभूष+ेत बुभूषेयाताम्=बुभूष+ेयाताम् बुभूषेरन्=बुभूष+ेरन्

AliN

बुभूषिषीष्ट =बुभूषि+षीष्ट बुभूषिषीयास्ताम्=बुभूषि+षीयास्ताम् बुभूषिषीरन्=बुभूषि+षीरन्

lriN

अबुभूषिष्यत=अबुभूषि+ ष्यत अबुभूषिष्येताम्=अबुभूषि+ष्येताम् अबुभूषिष्यन्त=अबुभूषि+ ष्यन्त

<u>भू यङन्त yaNanta कर्तृवाच्य active</u>

lat

बोभूयते=बोभूय+ ते बोभूयेते=बोभूय+ ेते बोभूयन्ते=बोभूय+ न्ते

lit

बोभूयाञ्चक्रे=बोभूय+ ाञ्चक्रे बोभूयाञ्चक्राते=बोभूय+ाञ्चक्राते बोभूयाञ्चक्रिरे=बोभूय+ ाञ्चक्रिरे

lut

बोभूयिता=बोभूयि+ ता बोभूयितारौ=बोभूयि+ तारौ बोभूयितारः=बोभूयि+ तारः

lrt

बोभूयिष्यते=बोभूयि+ष्यते वोभूयिष्येते=बोभूयि+ष्येते बोभूयिष्यन्ते=बोभूयि+षन्ते

lot

बोभूयताम्=बोभूय+ ताम् बोभूयेताम्=बोभूय+ेताम् बोभूयन्ताम्=बोभूय+न्ताम्

VliN

बोभूयेत=बोभूय+ बोभूयेयाताम्=बोभूय+ बोभूयेरन्=बोभूय+

<u>भू सन्नन्त sannanta कर्मवाच्य passive</u>

lat

बुभूष्यते=बुभूष्य+ते बुभूष्येते=बुभूष्य+ेते बुभूष्यन्ते=बुभूष्य+न्ते

lrt

बुभूषिष्यते= बुभूषि +ष्यते बुभूषिष्येते= बुभूषि +ष्येते बुभूषिष्यन्ते= बुभूषि +ष्यन्ते

lot

बुभूष्यताम्= बुभूष्य +ताम् बुभूष्येताम्= बुभूष्य +ेताम् बुभूष्यन्ताम्= बुभूष्य +न्ताम्

laN

अबुभूष्यत= अबुभूष्य +त अबुभूष्येताम्= अबुभूष्य +ेताम् अबुभूष्यन्त= अबुभूष्य + न्त

VliN

बुभूष्येत= बुभूष्य + ेत बुभूष्येयाताम्= बुभूष्य + ेयाताम् बुभूष्येरन् = बुभूष्य +ेरन्

AliN

बुभूषिषीष्ट = बुभूषि +षीष्ट बुभूषिषीयास्ताम्= बुभूषि+षीयास्ताम् बुभूषिषीरन्= बुभूषि +षीरन्

lriN

अबुभूषिष्यत=अबुभूषि +ष्यत अबुभूषिष्येताम्=अबुभूषि +ष्येताम् अबुभूषिष्यन्त=अबुभूषि + ष्यन्त

<u>भू णिजन्त nijanta कर्मवाच्य passive</u>

lat

भाव्यते = भाव्य +ते भाव्येते= भाव्य + ेते भाव्यन्ते= भाव्य +न्ते

lot

भाव्यताम्= भाव्य +ताम् भाव्येताम्= भाव्य +ेताम् भाव्यन्ताम्= भाव्य +न्ताम्

laN

अभाव्यत= अभाव्य +त अभाव्येताम्= अभाव्य +ेताम् अभाव्यन्त= अभाव्य +न्त

ViiN

भाव्येत= भाव्य + ेत भाव्येयाताम्= भाव्य +ेयाताम् भाव्येरन्= भाव्य +ेरन्

AliN

भाविषीष्ट=भावि+ षीष्ट भाविषीयास्ताम्=भावि+षीयास्ताम् भाविषीरन्=भावि+षीरन्

lriN

अभाविष्यत=अभावि+ अभाविष्येताम्=अभावि+ अभाविष्यन्त=अभावि+

भू यङन्त yananta कर्मवाच्य passive

lat

बोभूय्यते=बोभूय्य+ते बोभूय्येते=बोभूय्य+ेते बोभूय्यन्ते=बोभूय्य+न्ते

lot

बोभूय्यताम्=बोभूय्य+ताम् बोभूय्येताम्=बोभूय्य+ेताम् बोभूय्यन्ताम्=बोभूय्य+न्ताम्

laN

अबोभूय्यत=अबोभूय्य+त अबोभूय्येताम्=अबोभूय्य+ेताम् अबोभूय्यन्त=अबोभूय्य+न्त

VliN

बोभूय्येत=बोभूय्य+ेत बोभूय्येयाताम्=बोभूय्य+ेयाताम् बोभूय्येरन्=बोभूय्य+ेरन्

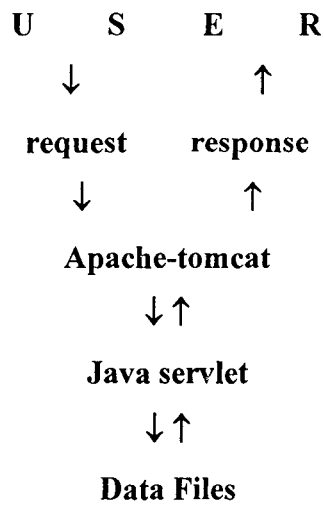
Chapter-4

Verb Analyzer for Sanskrit

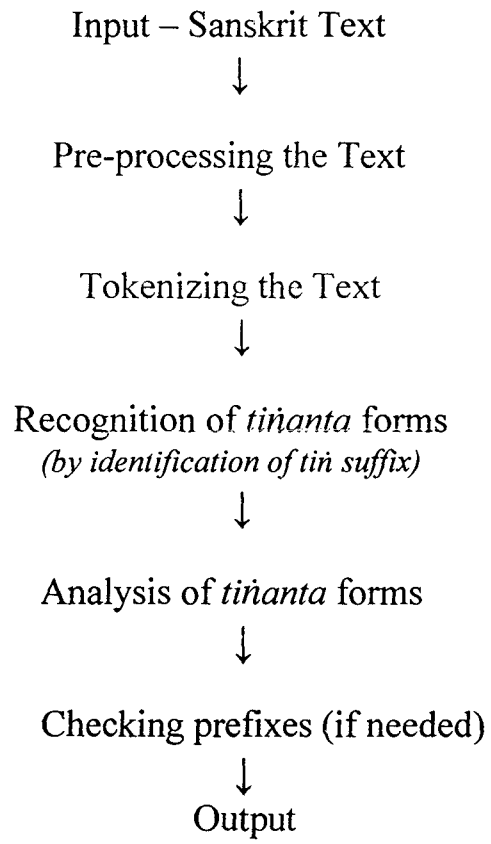
This chapter describes the partial implementation of Sanskrit verb analyzer as part of the present M. Phil. R & D. The morphological analysis methodology discussed in the previous chapter has been applied to develop a computational system which can identify and analyze Sanskrit verb forms. The computational model uses Java in the web format for the identification and analysis of regular Sanskrit verb-forms from Sanskrit texts according to Pāṇinīan and *Siddhānta Kaumudī (SK)* formalism. The system accepts words/sentences/ text Devanagari utf-8 input in the text area and gives analyzed output in the same format. The identification of *tiṅanta* verb forms depends on recognizing the *tiṅ* suffix or ending ir the words of the given text. The analysis strategy is based on separating the suffix from the base (and also prefix, if the case is so), then locating both of them in the respective lexical resources and then giving related information which is already stored in lexical files.

4.1 Architecture of the system

The following model describes the interaction between multi-tier architecture of the verb analyzer:



4.2 The Process-flow



Step I: Preprocessing

Preprocessing a text mainly consists of normalizing it. The text given as input may contain many irregular features such as numerals or English alphabets etc. due to typographical errors or other reasons.

Step II: Tokenization

Tokenization consists of separating out words from the running text. Tokenizer which takes the preprocessed text separates all the words of the text and returns them as separate tokens for further processing.

Step III: Identification of *tiṅanta* verb forms

The next task is to recognize the *tiṅanta padas* in the text which is already preprocessed and tokenized. The *tiṅanta* forms (forms which end in tiṅ suffix) are recognized by identifying the *tiṅ* suffixes or endings that remain at the end of Sanskrit verb forms. The

program, in this step, takes the help of the suffix list stored in the data files as back-end of the system.

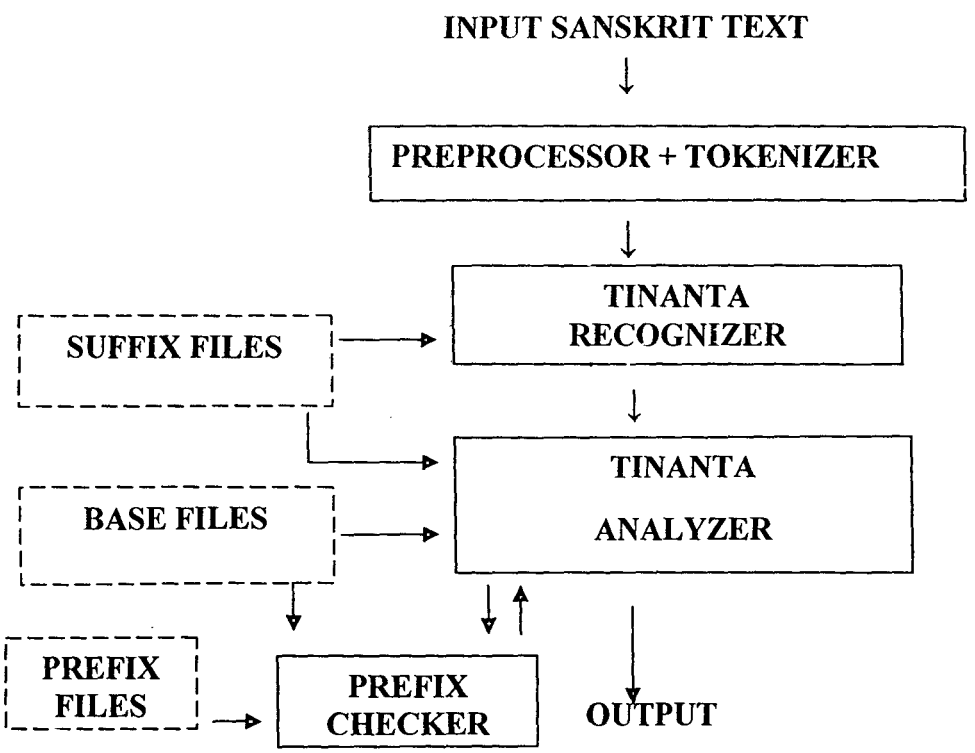
Step IV: Analysis of Sanskrit verb forms

The next step is to take all the identified *tiṅanta* forms for further analysis. The *tiṅ* ending has already been located in the third step. In this step the *tiṅ* ending is separated from the input word. The remaining string must be the verbal base, if no prefix is attached to it. This base is identified in the bases data-file which is a list of all possible bases of a root in various paradigms.

In case the base is prefixed with one or more upasargas, the system has to identify and separate prefix/es and then cut it to obtain the base. This is done with the help of a list of all the possible patterns of prefixes. The prefix thus identified is separated and we retrieve the verbal base which is matched in the base-list.

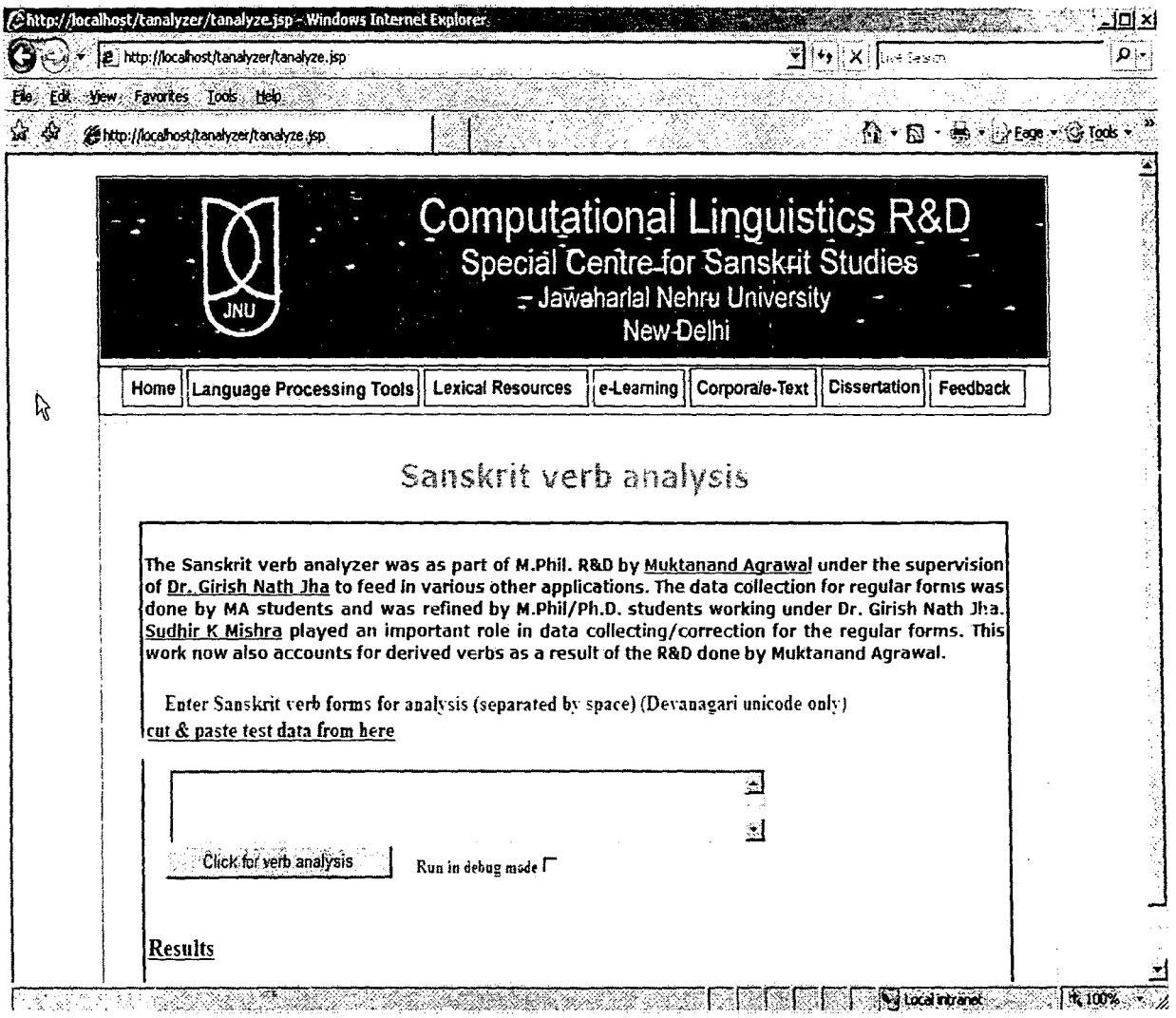
4.3 Module Description

The module description of the verb analyzer is given below -



4.3.1 The front-end: online interface

The front end of the system is the Graphical User Interface (GUI), visible to the users. It is live at http://sanskrit.jnu.ac.in . It has been created using JSP (Java Server Pages) and HTML components. The main JSP file *tanalyze.jsp* allows the user to feed the input in Devanagari utf-8 format using HTML text area component. Upon clicking the button labeled "Click for verb analysis" it calls the Java object "Verban" to process the input. The output returned by the Java objects is displayed to the user in Devanagari utf-8 format.



4.3.2The back-end: txt files

The back end contains lexical resources in the form of data-files. These are stored in simple text format in a way that the program can access it and retrieve related information.

The first one is the example-base which stores those verb forms which cannot be recognized and analyzed on the basis of present methodology. The forms like भव (loṭ, 2nd person, singular) which contain no sign of *tiṅ* suffix and have only the base remaining are stored here. The system checks these file in the very beginning so as to identify these forms before further analysis begins.

The second file is that of *tiṅ* endings which are stored in the text format with their relevant information. As indicated in previous chapters, the *tiṅ* endings contain the information of *lakāra* (tense/mood), person, number etc. Every *tiṅ* ending in the suffixes data-file is stored along with this information. The suffixes file is used in two steps. First, they are used in recognizing the *tiṅanta* forms by locating their *tiṅ* suffixes. Secondly, in the analysis step, the same *tiṅ* suffixes retrieve the information which is stored with every entry. The segregation of the ending from the *tiṅanta* verb from gives us the verbal base.

The third file contains the verbal bases of verb roots of *bhvādigaṇa*. The bases are arrived at in the analysis by separating the *tiṅ* ending from the *tiṅanta* verb form. In case a prefix is attached to the verb form, the string is not recognized as the base directly and has to undergo the prefix check process. The bases, in the files, are also stored with relevant information in the same manner as suffixes.

Another text file contains the Sanskrit prefixes. These are 22 in number but can form new patterns by getting attached or due to morphophonemic changes. The prefixes file contains all the prefixes with all possible patterns which can appear in Sanskrit. The prefixes, unlike previous two files, are not stored with any information as for now. In future, if it is needed, it can easily be added to them.

Given below is a sample of all txt files mentioned above to store data –

- examplebase.txt which stores verb forms which cannot be analyzed

भव=भू,कर्तृ,भ्वादि,परस्मै,लोट्,प्रथम,एकवचन,सिप्;बभूव=भू,कर्तृ,भ्वादि,परस्मै,लिट्,प्रथम,एकवचन,तिप्,_:

भू,कर्तृ,भ्वादि,परस्मै,लिट्,मध्यम,बहुवचन,थ:भू,कर्तृ,भ्वादि,परस्मै,लिट्,उत्तम,एकवचन,मिप्,_;

- bases.txt (which stores all possible verb bases of roots with the related information) in the following format

भव=भू,लट्,_,कर्तृवाच्य;बभूव=भू,लिट्,_,कर्तृवाच्य;भवि=भू,लुट्,_,कर्तृवाच्य;भवि=भू,लृट्,_,कर्तृवाच्य;भव=भू,लोट्,_,कर्तृवाच्य;अभव=भू,लङ्,_,कर्तृवाच्य;भव=भू,विधिलिङ्,_,कर्तृवाच्य;भू=भू,आशीर्लिङ्,_,कर्तृवाच्य;अभू=भू,लुङ्,_,कर्तृवाच्य;अभवि=भू,लृङ्,_,कर्तृवाच्य;भूय=भू,लट्,_,कर्मवाच्य;बभूव=भू,लिट्,_,कर्मवाच्य;भवि/भावि=भू,लुट्,_,कर्मवाच्य;भवि=भू,लृट्,_,कर्मवाच्य;भूय=भू,लोट्,_,कर्मवाच्य;अभूय=भू,लङ्,_,कर्मवाच्य;भूय=भू,विधिलिङ्,_,कर्मवाच्य;भवि/भावि=भू,आशीर्लिङ्,_,कर्मवाच्य;अभावि/अभवि=भू,लुङ्,_,कर्मवाच्य;अभवि=भू,लृङ्,_,कर्मवाच्य;बुभूष=भू,लट्,सन्नन्त,कर्तृवाच्य;बबभूव=भू,लिट्,सन्नन्त,कर्तृवाच्य;बुभूष्=भू,लुट्,सन्नन्त,कर्तृवाच्य;बुभूष्=भू,लृट्,सन्नन्त,कर्तृवाच्य;बुभूष्=भू,लोट्,सन्नन्त,कर्तृवाच्य;अबुभूष=भू,लङ्,सन्नन्त,कर्तृवाच्य;बुभूष्=भू,विधिलिङ्,सन्नन्त,कर्तृवाच्य;बुभूष्=भू,आशीर्लिङ्,सन्नन्त,कर्तृवाच्य;अबुभूष=भू,लुङ्,सन्नन्त,कर्तृवाच्य;अबुभूष=भू,लृङ्,सन्नन्त,कर्तृवाच्य;बुभूष्य=भू,लट्,सन्नन्त,कर्मवाच्य;बभूव=भू,लिट्,सन्नन्त,कर्मवाच्य;बुभूष्=भू,लुट्,सन्नन्त,कर्मवाच्य;बुभूष्=भू,लृट्,सन्नन्त,कर्मवाच्य;बुभूष्=भू,लोट्,सन्नन्त,कर्मवाच्य;अभव्=भू,लङ्,सन्नन्त,कर्मवाच्य;बुभूष्=भू,विधिलिङ्,सन्नन्त,कर्मवाच्य;बुभूष्=भू,आशीर्लिङ्,सन्नन्त,कर्मवाच्य;अबुभूष्=भू,लुङ्,सन्नन्त,कर्मवाच्य;अबुभूष्=भू,लृङ्,सन्नन्त,कर्मवाच्य;भावय=भू,लट्,णिजन्त,कर्तृवाच्य;भावय=भू,लिट्,णिजन्त,कर्तृवाच्य;भावयि=भू,लुट्,णिजन्त,कर्तृवाच्य;भावयि=भू,लृट्,णिजन्त,कर्तृवाच्य;भावय=भू,लोट्,णिजन्त,कर्तृवाच्य;अभावय=भू,लङ्,णिजन्त,कर्तृवाच्य;भावय=भू,विधिलिङ्,णिजन्त,कर्तृवाच्य;भाव्=भू,आशीर्लिङ्,णिजन्त,कर्तृवाच्य;अबीभव=भू,लुङ्,णिजन्त,कर्तृवाच्य;अभावयि=भू,लृङ्,णिजन्त,कर्तृवाच्य;भाव्य=भू,लट्,णिजन्त,कर्मवाच्य;भावय=भू,लिट्,णिजन्त,कर्मवाच्य;भावि=भू,लुट्,णिजन्त,कर्मवाच्य;भावि=भू,लृट्,णिजन्त,कर्मवाच्य;भाव्य=भू,लोट्,णिजन्त,कर्मवाच्य;अभाव्य=भू,लङ्,णिजन्त,कर्मवाच्य;भाव्य=भू,विधिलिङ्,णिजन्त,कर्मवाच्य;भावि=भू,आशीर्लिङ्,णिजन्त,कर्मवाच्य;अभावि=भू,लुङ्,णिजन्त,कर्मवाच्य;अभावि=भू,लृङ्,णिजन्त,कर्मवाच्य;बोभूय=भू,लट्,यङन्त,कर्तृवाच्य;बोभूय=भू,लिट्,यङन्त,कर्तृवाच्य;बोभयि=भू,लुट्,यङन्त,कर्तृवाच्य;बोभयि=भू,लृट्,यङन्त,कर्तृवाच्य;बोभूय=भू,लोट्,यङन्त,कर्तृवाच्य;अबोभूय=भू,लङ्,यङन्त,कर्तृवाच्य;बोभूय=भू,विधिलिङ्,यङन्त,कर्तृवाच्य;बोभूयि=भू,आशीर्लिङ्,यङन्त,कर्तृवाच्य;अबोभूयि=भू,लुङ्,यङन्त,कर्तृवाच्य;अबोभूयि=भू,लृङ्,यङन्त,कर्तृवाच्य;बोभूय्य=भू,लट्,यङन्त,कर्मवाच्य;बोभूय=भू,लिट्,यङन्त,कर्मवाच्य;बोभूयि=भू,लुट्,यङन्त,कर्मवाच्य;बोभूयि=भू,लृट्,यङन्त,कर्मवाच्य;बोभूय्य=भू,लोट्,यङन्त,कर्मवाच्य;अबोभूय्य=भू,लङ्,यङन्त,कर्मवाच्य;बोभूय्य=भू,विधिलिङ्,यङन्त,कर्मवाच्य;बोभूयि=भू,आशीर्लिङ्,यङन्त,कर्मवाच्य;अबोभूयि=भू,लुङ्,यङन्त,क

र्मवाच्य;अबोभूयि=भू,लृङ्,यङन्त,कर्मवाच्य;बोभो/बोभू/बोभवी=भू,लट्,यङ्लुगन्त,कर्तृवाच्य;बोभव=भू,लिट्,य ङ्लुगन्त,कर्तृवाच्य;बोभवि=भू,लुट्,यङ्लुगन्त,कर्तृवाच्य;बोभवि=भू,लृट्,यङ्लुगन्त,कर्तृवाच्य;बोभो/बोभू/बोभ वी=भू,लोट्,यङ्लुगन्त,कर्तृवाच्य;अबोभव्/अबोभू=भू,लङ्,यङ्लुगन्त,कर्तृवाच्य;बोभू=भू,विधिलिङ्,यङ्लुगन्त, कर्तृवाच्य;बोभू=भू,आशीर्लिङ्,यङ्लुगन्त,कर्तृवाच्य;अबोभव्/अबोभो=भू,लुङ्,यङ्लुगन्त,कर्तृवाच्य;अबोभवि=भू, लृङ्,यङ्लुगन्त,कर्तृवाच्य

- suffixes.txt (which stores *tin* suffixes along with their relevant information) in the following format

ति=लट्,प्रथम,एक,परस्मै;तः=लट्,प्रथम,द्वि,परस्मै;न्ति=लट्,प्रथम,बहु,परस्मै;सि=लट्,मध्यम,एक,परस्मै;षि= लट्,मध्यम,एक,परस्मै;थः=लट्,मध्यम,द्वि,परस्मै;थ=लट्,मध्यम,बहु,परस्मै;मि=लट्,उत्तम,एक,परस्मै;ामि= लट्,उत्तम,एक,परस्मै;वः=लट्,उत्तम,द्वि,परस्मै;ावः=लट्,उत्तम,द्वि,परस्मै;मः=लट्,उत्तम,बहु,परस्मै;ामः=लट्,उ त्तम,बहु,परस्मै;ाञ्चकार=लिट्,प्रथम,एक,परस्मै;तुः=लिट्,प्रथम,द्वि,परस्मै;ाञ्चक्रतुः=लिट्,प्रथम,द्वि,परस्मै;ु ः=लिट्,प्रथम,बहु,परस्मै;ाञ्चक्रुः=लिट्,प्रथम,बहु,परस्मै;िथ=लिट्,मध्यम,एक,परस्मै;ाञ्चकर्थ=लिट्,मध्य म,एक,परस्मै;थुः=लिट्,मध्यम,द्वि,परस्मै;ाञ्चक्रथुः=लिट्,मध्यम,द्वि,परस्मै;ाञ्चक्र=लिट्,मध्यम,बहु,परस्मै; ाञ्चकार=लिट्,उत्तम,एक,परस्मै;िव=लिट्,उत्तम,द्वि,परस्मै;ाञ्चकृव=लिट्,उत्तम,द्वि,परस्मै;िम=लिट्,उत्तम, बहु,परस्मै;ाञ्चकृम=लिट्,उत्तम,बहु,परस्मै;ता=लुट्,प्रथम,एक,परस्मै;तारौ=लुट्,प्रथम,द्वि,परस्मै;तारः=लुट्,प्रथ म,बहु,परस्मै;तासि=लुट्,मध्यम,एक,परस्मै;तास्थः=लुट्,मध्यम,द्वि,परस्मै;तास्थ=लुट्,मध्यम,बहु,परस्मै;ता स्मि=लुट्,उत्तम,एक,परस्मै;तास्वः=लुट्,उत्तम,द्वि,परस्मै;तास्मः=लुट्,उत्तम,बहु,परस्मै;ष्यति=लृट्,प्रथम,एक,पर स्मै;स्यति=लृट्,प्रथम,एक,परस्मै;ष्यतः=लृट्,प्रथम,द्वि,परस्मै;स्यतः=लृट्,प्रथम,द्वि,परस्मै;ष्यन्ति=लृट्,प्रथम,ब हु,परस्मै;स्यन्ति=लृट्,प्रथम,बहु,परस्मै;ष्यसि=लृट्,प्रथम,एक,परस्मै;स्यसि=लृट्,प्रथम,एक,परस्मै;ष्यथः=लृट्, प्रथम,द्वि,परस्मै;स्यथः=लृट्,प्रथम,द्वि,परस्मै;ष्यथ=लृट्,प्रथम,बहु,परस्मै;स्यथ=लृट्,प्रथम,बहु,परस्मै;ष्यामि=लृ ट्,प्रथम,एक,परस्मै;स्यामि=लृट्,प्रथम,एक,परस्मै;ष्यावः=लृट्,प्रथम,द्वि,परस्मै;स्यावः=लृट्,प्रथम,द्वि,परस्मै;ष्या मः=लृट्,प्रथम,बहु,परस्मै;स्यामः=लृट्,प्रथम,बहु,परस्मै;तु=लोट्,प्रथम,एक,परस्मै;तात्=लोट्,प्रथम,एक,परस्मै; ताम्=लोट्,प्रथम,द्वि,परस्मै;न्तु=लोट्,प्रथम,बहु,परस्मै;तम्=लोट्,मध्यम,द्वि,परस्मै;त=लोट्,मध्यम,बहु,परस्मै; ानि=लोट्,उत्तम,एक,परस्मै;ाव=लोट्,उत्तम,द्वि,परस्मै;ाम=लोट्,उत्तम,बहु,परस्मै;त्=लङ्,प्रथम,एक,परस्मै;ता म्=लङ्,प्रथम,द्वि,परस्मै;न्=लङ्,प्रथम,बहु,परस्मै;ः=लङ्,मध्यम,एक,परस्मै;तम्=लङ्,मध्यम,द्वि,परस्मै;त=लङ् ,मध्यम,बहु,परस्मै;म्=लङ्,उत्तम,एक,परस्मै;ाव=लङ्,उत्तम,द्वि,परस्मै;ाम=लङ्,उत्तम,बहु,परस्मै;ेत्=विधिलि ङ्,प्रथम,एक,परस्मै;ेताम्=विधिलिङ्,प्रथम,द्वि,परस्मै;ेयुः=विधिलिङ्,प्रथम,बहु,परस्मै;ेः=विधिलिङ्,मध्यम ,एक,परस्मै;ेतम्=विधिलिङ्,मध्यम,द्वि,परस्मै;ेत=विधिलिङ्,मध्यम,बहु,परस्मै;ेयम्=विधिलिङ्,उत्तम,एक,प रस्मै;ेव=विधिलिङ्,उत्तम,द्वि,परस्मै;ेम=विधिलिङ्,उत्तम,बहु,परस्मै;यात्=आशीर्लिङ्,प्रथम,एक,परस्मै;यास्ता म्=आशीर्लिङ्,प्रथम,द्वि,परस्मै;यासुः=आशीर्लिङ्,प्रथम,बहु,परस्मै;याः=आशीर्लिङ्,मध्यम,एक,परस्मै;यास्तम्= आशीर्लिङ्,मध्यम,द्वि,परस्मै;यास्त=आशीर्लिङ्,मध्यम,बहु,परस्मै;यासम्=आशीर्लिङ्,उत्तम,एक,परस्मै;यास्व= आशीर्लिङ्,उत्तम,द्वि,परस्मै;यास्म=आशीर्लिङ्,उत्तम,बहु,परस्मै;त्=लुङ्,प्रथम,एक,परस्मै;ीत्=लुङ्,प्रथम,एक,पर स्मै;ताम्=लुङ्,प्रथम,द्वि,परस्मै;िष्टाम्=लुङ्,प्रथम,द्वि,परस्मै;वन्=लुङ्,प्रथम,बहु,परस्मै;िषुः=लुङ्,प्रथम,बहु,पर स्मै;ः=लुङ्,मध्यम,एक,परस्मै;ीः=लुङ्,मध्यम,एक,परस्मै;तम्=लुङ्,मध्यम,द्वि,परस्मै;िष्टम्=लुङ्,मध्यम,

द्वि,परस्मै;त=लुङ्,मध्यम,बहु,परस्मै;ि=लुङ्,मध्यम,बहु,परस्मै;वम्=लुङ्,उत्तम,एक,परस्मै;िषम्=लुङ्,उत्तम,एक,परस्मै;व=लुङ्,उत्तम,द्वि,परस्मै;ि्व=लुङ्,उत्तम,द्वि,परस्मै;म=लुङ्,उत्तम,बहु,परस्मै;ि्म=लुङ्,उत्तम,बहु,परस्मै;

Prefixes.txt

अति;अधि;अनु;अन्तर्;अप;अपि;अभि;अव;आ;उत्;उद्;उप;दुर्;नि;निर्;परा;परि;प्र;प्रति;वि;सं;सम्;सु;प्रवि;व्यव;अभिनि;निरव;अपा;अभ्या;उदा;अध्या;सम्परि;व्या;प्रणि;

4.3.3 The web server

The verb analyzer runs on Apache Tomcat 4.0 platform. The details for this Java based webserver follows -

4.3.3.1Apache Tomcat 4.0

Apache Tomcat is the servlet container that is used for the Java Servlet and JavaServer Pages technologies. The Java Servlet and Java Server Pages specifications are developed by Sun under the Java Community Process. Apache Tomcat is developed in an open and participatory environment and released under the Apache Software License. Apache Tomcat is intended to be a collaboration of the best-of-breed developers from around the world[1].

4.3.3.2 Java Servlet Technology

Java Servlet technology provides web developers with a simple, consistent mechanism for extending the functionality of a web server and for accessing existing business systems. A servlet can almost be thought of as an applet that runs on the server side--without a face. Java servlets make many web applications possible[2].

4.3.3.3 Java Server Pages

Java Server Pages (JSP) technology provides a simplified, fast way to create dynamic web content. JSP technology enables rapid development of web-based applications that are server and platform-independent[3]. JSP pages are, however, compiled into servlets. Still, it is better to use JSP pages instead of always using servlets because JSP technology

[1] Apache Tomcat website, http://www.apache.org/
[2] http://java.sun.com/products/servlet/
[3] http://java.sun.com/products/jsp/

separates the web-presentation from the web-content and thus simplifies the process of creating pages. Basically JSP pages use XML tags and scriptlets written in the Java programming language to encapsulate the logic that generates the content for the web page. On the other hand, it passes any formatting (HTML or XML) tags directly back to the response page. In this way, JSP pages separate the page logic from its design and display. It is one of the most sophisticated tools available for high performance and secures web applications.

4.4 Main class: Verban

This is the main class of the program.

```
public class Verban{

}
```

It tokenizes the input text, gets it preprocessed, gets *tiṅantas* identified and then analyze the *tiṅanta padas* with the help of the lexical resources. Finally, this module displays the results. This class has following methods –

```
String preProcess(String txt)
public String tagVerb(String txt)
private String analyzeDerivedVerbs(String verb)
public String printErr()
```

4.4.1Preprocessor

This module first normalizes the input and then checks if there are any irregularities or typographical errors.

```
String preProcess(String txt){

        if (txt.length() > 0){
              txt = txt.replace('"','\'');
              txt = txt.replace('\n',' ');
        }

        return txt;

}
```

4.4.2 Tokenizer

Tokenization segregates all the word forms and presents them one by one for further processing. For tokenization of data, the program uses StringTokenizer class of Java.

```
StringTokenizer verbData = new StringTokenizer(txt," ");
while (verbData.hasMoreTokens()){

      aVerb=verbData.nextToken().trim();
      }
```

4.4.3*Tiṅanta* identifier

The first task of the system, after tokenizing the words is to identify *tiṅ* endings in the verb forms. A text will consist of various categories of words. The *tiṅanta* analyzer will have to take care only of *tiṅanta* verb forms. So, the recognition of *tiṅanta* forms is of primary importance. Sample of this function is given below-

```
      String tkn = "";
      String suffix = "";
      String base = "";
      String suffixTag = "";
      String baseTag = "";

      if(tkn.indexOf("=")>0){

      suffix = tkn.substring(0,tkn.indexOf("="));  //suf

      suffixTag = tkn.substring(tkn.indexOf("=")+1,tkn.length());
//the suff tag
                  }
      if ( verb.lastIndexOf(suffix) > 0 ){
      base = verb.substring(0,verb.lastIndexOf(suffix));
//un-confirmed base
                        break;
                  }
```

4.4.4*Tiṅanta* Analyzer

After identifying the suffixes, and tiṅanta thereof, the next step is to analyze tiṅanta forms. The analysis is done by following object:

```
private String analyzeDerivedVerbs(String verb)
```

This object has following separate methods to accomplish this task:

Identification of the base

```
if (base.length()>0){
            st = new StringTokenizer(bases.toString(), ";");
            String tmpBase ="";
            while (st.hasMoreTokens()){
            tkn = st.nextToken();
if(tkn.indexOf("=")>0){
tmpBase = tkn.substring(0,tkn.indexOf("="));
                        }
if (base.equals(tmpBase)){
baseTag = tkn.substring(tkn.indexOf("=")+1,tkn.length());
                        break;
                        }
                }
            }

if (base.length()>0 && baseTag.length()==0) {
                //check it in base database
st = new StringTokenizer(bases.toString(), ";");
            String tmpBase ="";
            String tmpTkn = "";
if(tmpTkn.indexOf("=")>0){
tmpBase = tmpTkn.substring(0,tmpTkn.indexOf("="));  //base from
dict
```

```
                    }
if (base.equals(tmpBase)){

baseTag =tmpTkn.substring(tmpTkn.indexOf("=")+1,tmpTkn.length());
//the base tag
                    \
                      break;
                      }
                  }//while
     }//if
```

Identification of prefixes

```
String prefix ="";

     if (base.length()>0 && baseTag.length()==0) {

          st = new StringTokenizer(prefixes.toString(), ";");

     while (st.hasMoreTokens()){

          prefix = st.nextToken().trim();

     if (base.indexOf(prefix)==0){

          base = base.substring(prefix.length(), base.length());

          break;
                    }
            }
   }
```

4.5 Test corpora

The corpus for testing the system is consisted up of verb forms of Sanskrit verb roots which can be accessed by clicking the link 'cut & paste data from here' above the

textarea field on the same page. The data can also be acquired by using the tiṅanta generator on the same website. The generator produces verb forms in different paradigms for selected verb/s. To check the system, one can copy this generated data (which is in UTF-8 devanagari format) and paste in the text-area field of analysis page. Another form of giving input is simply to type the data directly in the textarea in UTF-8 devanāgarī format using a Unicode IME like Baraha.

4.6 How it works

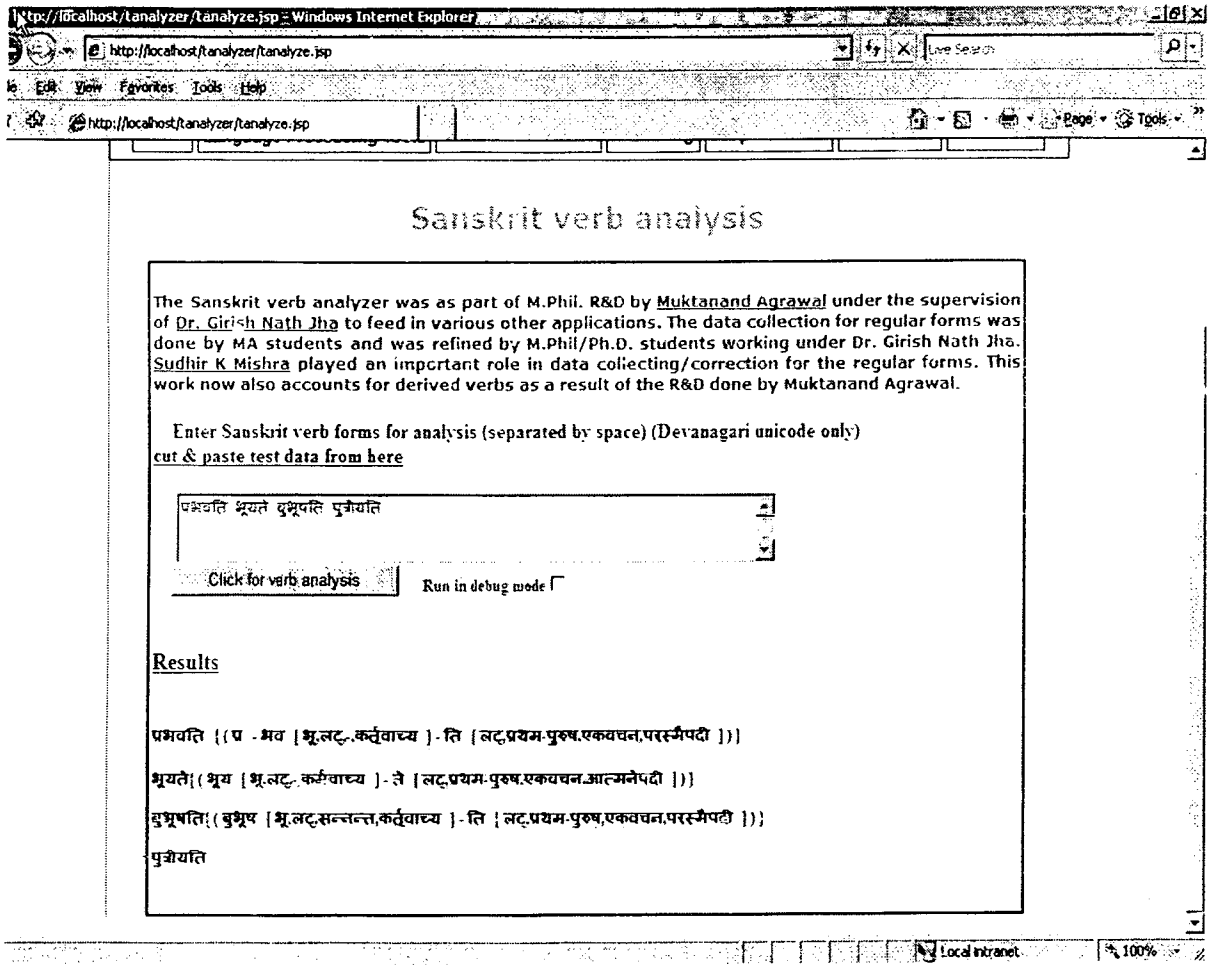
On the localhost (CD version), the website can be opened by the URL http://localhost:8080/verbs/analyze.jsp. On the actual server, the URL is http://www.sanskrit.jnu.ac.in/tanalyze.jsp . The home page of the site has already been given in this chapter.

The site accepts devanagari data in utf-8 format. Therefore, a Unicode IME like Baraha[4] has to be installed. Otherwise, user can enter some the test files provide.

Upon clicking the button labeled "Click here for verb analysis" the JSP interface sends data to the Verban object, which after preprocessing and tokenizing the input sends each word to the java object for analysis. The object keeps on building the display depending on the output from the proproessor-recognizer and analyzer objects.

The next screen shot illustrates some analysis of data input which is explained in the next section.

[4] http://www.baraha.com/BarahaIME.htm



4.7 Input-Output examples -

Input text

Given below is a sample input data containing various types of verb forms.

प्रभवति भूयते बुभूषति पुत्रीयति

Output text

The analysis will be as follows as is shown on the screen-shot given above:

प्रभवति = प्र –भव [भू, लट्, _, कर्तृवाच्य] – ति [लट्, प्रथम-पुरुष, एकवचन, परस्मैपदी]

भूयते = भूय [भू, लट्, _, कर्मवाच्य] – ते [लट्, प्रथम-पुरुष, एकवचन, आत्मनेपदी]

बुभूषति = बुभूष [भू, लट्, सन्नन्त, कर्तृवाच्य] – ति [लट्, प्रथम-पुरुष, एकवचन, परस्मैपदी]

पुत्रीयति no result is given.

Conclusion

Creating tools for morphological analysis of a language requires a lot of data as well as rules. The analysis model has to take care of all the features of morphology of the given language. The present work which is an R&D for building a morphological analyzer for Sanskrit verb forms of *bhvādigaṇa* is a step towards developing a full-fledged strategy for comprehensive analysis of Sanskrit (http://sanskrit.jnu.ac.in). Similar work for noun-words analysis, POS tagging, gender analyzer and so on have already been undertaken by the same team and a considerable progress have been made. Still, Sanskrit is very highly developed in morphology. The words are very complex, though the processes of inflection and derivation are very well defined step-by-step in the grammatical analysis. Therefore, to develop a morph analyzer is a challenging task. Given below are the limitations and future extension scope of the system.

Limitations:

1. The present analysis is confined to the analysis of the roots of *bhvādigaṇa.* Though this class contains around half of the total primitive verb roots of Sanskrit most of them being used popularly, the pattern followed throughout the class is very simple and does not deviate much from the general archetype. The three other conjugational classes 4^{th}, 6^{th} and 10^{th} also accept a similar and simple methodology to form the verb forms. The case is, however, different with rest of the 6 non-conjugational classes. These classes show a pattern which is quite different from previous four classes. In fact, these classes not only vary among themselves but in a single class itself, the roots adopt different patterns. Therefore, extending the analysis model to all the 10 classes of DP is not a simple task.

   However, given the fact that unlike regular verb forms, the derived verb forms including passives and impersonals follow a rather similar pattern across different classes, not taking care of the class they belong to. So, less problems

are expected to appear if system is extended further on a similar path for derived verbs.

2. The model described in this work is based on realization of morphemes in a word, i.e. identifying the tiṅ endings and bases of a verb form. The primary and most important function is the recognition of *tiṅ* ending. Locating the *tiṅ* ending in a word confirms it to be a *tiṅanta* form on one hand; on other hand, separating the tiṅ affix produces the verbal base. So, *tiṅ* endings are the key in this analysis module. The verb forms which do not contain any tiṅ termination in the end are problematic to be recognized and analyzed. For example bhava and babhūva are such forms. As for now, they are being stored in example base.

3. The strategy is unable to differentiate among the various types of verb forms which are similar. We have seen that their may occur similar surface realizations for different lexical combinations. For example the forms for 1st person in luṅ lakāra parasmaipada are quite similar to that of ātmanepada. Present strategy cannot differentiate between them because it only analyzes morphemes.

   Moreover, if a surface realization like bhavati, which can be verb form and at the same time is also be a noun or participle cannot be disambiguated through this system.

Future Extensions and Implementations:

The present system confines itself to the analysis of bhvādigaṇa. The analysis model can be extended to the remaining nine classes of DP. While making the rules for derived verb forms is simpler due to similarity

The system is also expected to accumulate the denominative verbal forms of Sanskrit, i.e. nāmadhātus. They are similar to other verb forms in the sense they accept same tiṅ

terminations and are declined in the same way. However they involve a rather different derivational process. The two components needed to extend the system for nāmadhātu analysis are derivational rules and a lexicon of nominal words of Sanskrit. The derivational rules will help to identify the nominal stem from which the root is derived. The lexicon will be checked to confirm the retrieved nominal stem.

Kṛdantas are also commonly use in Sanskrit to denote the sense of verb. The analyzer, to become a full-fledged analyzer for Sanskrit verb-forms of any kind, will have to develop a strategy for analyzing them.

Bibiliography

Books

- Sharma, Rama Nath, 2003, 'The Aṣṭādhyāyi of Pāṇini', Munshiram Manoharlal Publishers Pvt. Ltd., Delhi.
- Shastri Dwarikadas (ed.), 2000, "Mādhavīyā Dhātuvṛttiḥ", Tara Book Agency, Varanasi.
- Jijñāsu, Pt. Brahmadatta, 2003, 'Aṣṭādhyāyi Bhāṣya Vṛtti' Ramlal Kapoor Trust, Hariyana.
- Upadhye, P.V. , 1927, "Dhatupupachandrika," Gopal Narayan & Co., Bombay.
- Whitney, William Dwight, 1983, 'The Root verb-forms and Primary derivatives of the Sanskrit language' Motilal Banarasidass, Delhi.
- Whitney, W.D., 2002, 'History of Sanskrit grammar' Sanjay prakashan, Delhi.
- Whitney, William Dwight, 1983, 'Sanskrit Grammar' Motilal Banarasidass, Delhi.
- Williams, Monier, "A Practical Grammar of the Sanskrit Language", Clarendon Press, Oxford.
- Abbi, Anvita. Semantic universals in Indian language, Indian Institute of Advance Study,Shimla,1994
- Bhandarkar, R.G., 1924, 'First book of Sanskrit' Radhabai Atmaram Sagoon, Bombay.
- Bhandarkar, R.G., 1924, 'Second book of Sanskrit' Radhabai Atmaram Sagoon, Bombay.

- Wilson, H. H., 1841, "An Introduction to the Grammar of the Sanskrit Language", J. Madden & Co., London.
- Kapoor Kapil, 2005, "Dimensions of Pāṇini Grammar: the Indian Grammatical System", D.K. Printworld (P) Ltd., New Delhi.
- Balagurusamy, E., 1998, "Programming with Java: A Primer", Tata McGraw-Hill Publishing Company Limited, New Delhi.
- Misra, Vidya Niwas, 1966, "The Descriptive Technique of Pāṇini", Mouton & Co. The Hague, Paris.
- Kapoor, Kapil, 1985, Semantic Structures and the Verb: A propositional analysis (Intellectual Publications, New Delhi, 1985).
- Jurafsky, Daniel and Martin H. James, Speech and languages processing, Prentice Hall, India
- Kale, M.R., 1972, 'A Higher Sanskrit grammar' Motilal Banarasidass, Delhi.
- Katre, S.M., 1968, 'Dictionary of Pāṇini' Deccan College, Poona.
- Joshi, Sivaram Dattatray & Roodbergen J. A. F., 1998, The Aṣṭādhyāyi of Pāṇini', Sahitya Akademi, New Delhi.
- Bharati, Akshar, Vineet Chaitanya and Rajeev Sangal, 1995, Natural Language Processing: A Paninian Perspective, Prentice-Hall of India, New Delhi.
- Cardona, George, Pāṇini: His work and its traditions, Motilalal Banarasidass, New Delhi
- Muller, F. Max, 1983, 'A Sanskrit grammar' Asian Educational Services, Delhi.
- Shastri Charu Deva, 1991, 'Pāṇini : Re-interpreted' Motilal Banarasidass, Delhi.
- Anderson, An Introduction to Natural languages processing, Prentice Hall India
- Giridharsharma Chaturvedi (ed.), 2004, Vaiyakaraṇasiddhāntakaumudī with Bālamanorama and Tattvabodhini ṭikā, Motilal Banarasidass, Delhi.
- Hart, George L., 1984, 'A Rapid Sanskrit method' Motilal Banarasidass, Delhi.

- Joshi, Sivaram Dattatray & Roodbergen J. A. F., 1975, 'Patañjali's Vyākarana Mahābhāsya, Kārakāhnika (P. 1.4.23-1.4.55)' CASS, University of Poona, Poona.
- Kielhorn, F., 1970, 'Grammar for Sanskrit Language' Chowkhamba Sanskrit Series office, Varanasi.
- Macdonell, A. A., 1997 (Reprint), 'A Sanskrit Grammar for Students', D. K. Printworld (P) Ltd., New Delhi.
- Narayan Mishra(ed). Kashika of Pt.Vamana and Jayaditya ,Chaukhamba Sanskrit sansthan, Varanasi,1996.
- Nautiyal, Chakradhar Hans, 1995, 'Bṛhada-anuvāda-candrikā', Motilal Banarasidass, Delhi.
- Pt. Brahmadatta Jijñāsu (ed.), 1998, 'Panini-Astadhyayi', Ramlal Kapoor Trust, Sonipat.
- Yogi, Satyabhushana & Shashikumar, 1985, 'Nighaṇṭu tathā nirukta', Motilal Banarasidass, Delhi.
- Sharma, Ramlakhan and Acharya Sudhir Tiwari, 1982, "Sanskrit-dhātu-saraṇiḥ", Ram Prasad & Brothers, Etawah.
- Pandey, Gopaldutta (ed.), 2003 (Reprint), "Vaiyākaraṇa-siddhāntakaumudī", Caukhambhā Surabhāratī Prakāśana.
- Palsule, Gajanan Balkrishna, 1961, "The Sanskrit Dhātupāṭhas: A Critical Study", Univ. of Poona, Poona.

Articles

- Nooten, B. A. van, Dec. 1967, "Pāṇini's Replacement Technique and the Active Finite Verb", Language (currently published by Linguistic Society of America), Vol. 43, No. 4. (Dec., 1967), pp. 883-902.
- Avery, John, "Contributions to the History of Verb-Inflection in Sanskrit",
- Journal of the American Oriental Society (currently published by American Oriental Society),Vol. 10. (1872 - 1880), pp. 219-324.

- B. Syamalakumari "On Defining Verb", Language In India, Vol. 3: 2 Feb. 2003
- Vijayanand Kommaluri, R. Subramanian, and Anand Sagar K "Issues in Morphological Analysis of North-East Indian Languages", Language In India, Volume 5: 7 July 2005.
- Subramanian, Aparna, "Sanskrit to English Translator", Language In India, Volume 5: 1 January 2005.
- Jha Girish N, 1994, Indian theory of knowledge: an AI perspective (proc. of national seminar on "Interface Mechanisms in Shastras and Computer Science", Academy of Sanskrit Research, Melkote, Mysore, April, 1994)
- 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", January, 1995)
- Buhler, Georg, 1985, The Roots of the Dhatupatha not found in Literature, A Reader on the Sanskrit grammarians (J.F.Stall ed.), Motilal Banarasidass, Delhi.
- Academy of Sanskrit Research, (ASR) Melkote,
- Cardona, George and Dhanesh Jain (eds.), 2003, "The Indo-Aryan Languages", Routledge Language Family Series, vol.2, Routledge: London and New York.
- Bharati Akshar, Amba P. Kulkarni Vineet Chaitanya, 1996, Challenges in developing word analyzers for Indian languages, Presented at Workshop on Morphology, CIEFL, Hyderabad.
- Bharati, , A., Sangal R., 1990, "A karaka based approach to parsing of Indian languages", proc of the 13 th COLING vol 3, pp 30-35, Finland.
- Bharati, Akshar and Rajeev Sangal, Parsing free wordorder languages using the Paninian framework, In ACL93: Proc.of Annual Meeting of Association for

Computational Linguistics, Association for Computational Linguistics, New York, 1993a.

- Bharati, Akshar, Vineet Chaitanya and Rajeev Sangal, A computational framework for Indian languages, Technical Report TRCS-90-100, Dept. of CSE, IIT Kanpur, July 1990b. (Course Notes for Intensive Course on NLP for Linguists, Vol.1)
- Cardona, George, 1967, 'Pāṇini's syntactic categories', Journal of the Oriental Institute, Baroda (JOIB) 16: 201-15.
- Cardona, George, 2004, 'Some Questions on Pāṇini's Derivational system' In SPLASH proc. of iSTRANS, pp. 3.
- Edgren, A. Hjalmar, 1885, 'On the verbal roots of the Sanskrit language and of the Sanskrit grammarians' Journal of the Americal Oriental Society 11: 1-55.
- G.V. Singh, Girish Nath Jha, Indian theory of knowledge: an AI perspective proc. of seminar, ASR, Melkote, Mysore, 1994
- Huet, Gerard, "Towards Computational Processing of Sanskrit",
- Bhate, Saroja and Subhash Kak, 1993, "Pāṇini's Grammar and Computer Science" Annals of the Bhandarkar Oriental Research Institute, vol. 72, pp. 79-94.
- Jha Girish Nath, Mishra, S K, Chandrashekar R, Subash, August, 2005, developing a Sanskrit Analysis System for Machine Translation presented, at the National Seminar on Translation Today: state and issues, Deptt. of Linguistics, University of Kerala, Trivandrum.
- Jha Girish Nath, October 2003, "A Prolog Analyzer/Generator for Sanskrit Subanta Padas", Language in India, Volume 3: 11.

- Jha Girish Nath, November, 2005 Language Technology in India: A survey Issue of C.S.I. magazine
- Jha Girish Nath, February 2004, The System of Panini, Language in India, volume 4:2
- Jha Girish Nath, March 2004, Generating nominal inflectional morphology in Sanskrit SIMPLE 04, IIT-Kharagpur Lecture Compendium, Shyama Printing Works, Kharagpur, WB,
- Jha, Girish Nath, December, 2003, Current trends in Indian languages technology, Langauge In India, Volume December.
- Joshi, S. D., 1962, 'Verbs and nouns in Sanskrit' Indian linguistics 32: 60-63.
- Kapoor, Kapil, 1996. Panini's derivation system as a processing model (to appear in the proc. of "A Symposium on Machine Aids for Translation and Communication, 11-12 April, School of Computer & Systems Sciences, J.N.U. New Delhi, 1996)
- Kiparsky, P. and Stall, J. F., 1969, "Syntactic and Semantic Relation in Panini" (Foundations of Language, Vol.5, 83-117).
- Kumar Sachin and Jha Girish Nath, December 2005, "A Paninian Sandhi analyzer for Sanskrit", in the proc. at Platinum Jubilee International Conference, L.S.I. at Hyderabad University, Hyderabad page-35
- Mishra Sudhir K, Jha Girish Nath, 2004, A karaka analyzer for Sanskrit Tata McGraw Hill, 2004, Proc of International Conference on Speech and Language Technology, New Delhi

- Mishra Sudhir K, Jha Girish Nath, 2005, Identifying verb inflections in Sanskrit morphology Proc. of SIMPLE05, IIT Kharagpur
- R.M.K. Sinha, 1989, "A Sanskrit based Word-expert model for machine translation among Indian languages", Proc. of workshop on Computer Processing of Asian Languages, Asian Institute of Technology, Bangkok, Thailand, Sept.26-28, 1989, pp. 82-91.12.
- Ramakrishnamacharyulu, K.V., Paninian Linguistics and Computational Linguistics, Samvit, Series no. 27. Pp. 52-62, Academy of Sanskrit Research, Melkote, Karnataka (India), 1993.
- Whitney, W.D., 1885, "The Roots of the Sanskrit Language", Transactions of the American Philological Association.

Theses/Dissertations:

- Jha Girish N, 1993, Morphology of Sanskrit Case Affixes, A Computational analysis Dissertation of M.Phil submitted to Jawaharlal Nehru University, New Delhi-110067.
- Chandrashekhara, R, 2006, 'POS Tagging for Sanskrit', submitted for Ph.D degree at SCSS, JNU.
- Chandra, Subash, 2006, 'Machine Recognition and Morphological Analysis of Subanta-padas', submitted for M.Phil degree at SCSS, JNU
- Mishra, Sudhir Kumar, 2007, 'Sanskrit Karaka Analyzer for Machine Translation', submitted for Ph.D. degree at SCSS, JNU

Web Sources

- The Web server, Apache Tomcat, http://www.apache.org/ (accessed: 12 July 2007).
- TDIL, http://tdil.mit.gov.in/mat/ach-mat.htm (accessed: 10 May 2007).
- Shabdabodha, TDIL, Gov. of India, http://tdil.mit.gov.in/download/Shabdabodha.html (accessed: 22 June 2007).
- RCILTS, School of Computer & System Science, http://rcilts.jnu.ac.in JNU, New Delhi. (Accessed: 17 Nov. 2006).
- RCILTS, Utkal University, http://www.ilts-utkal.org/nlppage.htm (accessed : June 30, 2007).
- Sanskrit academy http://www.sanskritacademy.org/Achievements.htm (accessed: June 30, 2007).
- Pen to Paper, The House Magazine of C-DAC, http://www.cdacindia.com/html/adp/mactrans.asp (accessed: June 07, 2007).
- Pen to Paper, The House Magazine of C-DAC, http://www.cdacindia.com/html/connect/3q2000/art10a.htm# (accessed: June 07, 2007).
- Peter M. Scharf and Malcolm D. Hyman, http://sanskritlibrary.org/morph/ (accessed: June 15, 2007).
- Peter M. Scharf and Malcolm D. Hyman, http://sanskritlibrary.org/ (accessed: June 15, 2007).
- http://www.cse.iitk.ac.in/users/langtech/anubharti.htm (accessed: June 15 2007).

- http://www-asia.human.is.tohoku.ac.jp/demo/vasia/html/ (accessed: Jan 15 2007).
- Huet, Gerard, http://sanskrit.inria.fr/ (accessed: June 30, 2007).
- Java Server Pages, http://java.sun.com/products/jsp/ (accessed: July 10, 2007).
- JAVA, Servlet, http://java.sun.com/products/servlet/ (accessed: July 10, 2007).
- Desika, http://tdil.mit.gov.in/download/Desika.htm (accessed: June 02, 2007).
- Anusaaraka, http://www.iiit.net/ltrc/Anusaaraka/anu_home.html (accessed: June 10, 2007).
- Baraha, Software, http://www.baraha.com/BarahaIME.htm (accessed: July 15 2007).
- Briggs, Rick, 1985, Knowledge representation in Sanskrit, *AI magazine*. Dr. Shivamurthy Swami, http://www.taralabalu.org/panini/ (accessed: 30 April 2007).