Computer Processing of Sanskrit
Nominal Inflections

Computer Processing of Sanskrit Nominal Inflections: Methods and Implementation

By

Subhash Chandra and Girish Nath Jha

CAMBRIDGE SCHOLARS PUBLISHING

Computer Processing of Sanskrit Nominal Inflections:
Methods and Implementation,
by Subhash Chandra and Girish Nath Jha

This book first published 2011

Cambridge Scholars Publishing

12 Back Chapman Street, Newcastle upon Tyne, NE6 2XX, UK

British Library Cataloguing in Publication Data
A catalogue record for this book is available from the British Library

Copyright © 2011 by Subhash Chandra and Girish Nath Jha

All rights for this book reserved. No part of this book may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior permission of the copyright owner.

ISBN (10): 1-4438-2684-7, ISBN (13): 978-1-4438-2684-6

To

My

Loving

Mom

And Late Dad

शब्दस्य परिणामोयमित्याम्नायविदो विदुः ।
छन्दोभ्य एव प्रथममेतद विश्वं व्यवर्तत ॥
एकस्य सर्वबीजस्य यस्य चेयमनेकधा ।
भोक्तृभोक्तव्यरूपेण भोगरूपेण च स्थिति ॥

[*śabdasya pariṇāmoyamityāmnāyavido viduḥ ।*
chandobhya eva prathamametada viśvaṃ vyavartata ॥
ekasya sarvabījasya yasya ceyamanekadhā ।
bhoktṛbhoktavyarūpeṇa bhogarūpeṇa ca sthiti ॥]

TABLE OF CONTENTS

PREFACE

The book entitled "Computer Processing of Sanskrit Nominal Inflections: Methods and Implementations" is a result of a Research and Development (R&D) at the Master of Philosophy (M. Phil.) level from Special Centre for Sanskrit Studies (SCSS), Jawaharlal Nehru University, New Delhi-110067 during 2004-2006 under the supervision of Dr. Girish Nath Jha. The title of the dissertation was *Machine Recognition and Morphological Analysis of subanta-padas*. More significant is the fact that this was the first dissertation in the newly established area of R&D at the Special Center for Sanskrit Studies after Dr. Jha joined the department in 2002. Details of this R&D group at JNU can be seen at:
http://sanskrit.jnu.ac.in

This work is based on the reverse engineering implementation of Pāṇini's Sanskrit Grammar. Pāṇini, the great Sanskrit grammarian of 7th BCE, has described Sanskrit in only 4000 rules (Jha 2004). Morphological analyses are key to Sanskrit analysis- a language extremely compact due to its rich morphology. On the surface level, Panini, like his predecessors, seems to have defined rules in a forward looking generative fashion which makes reverse analysis necessary for parsing. At times, the method followed in this R&D looks *ad-hoc* and un-Pāṇinian, but in reality it derives from the Pāṇini's formulations. Since parsing inflections is the first basic step towards complete analysis, the present work has relevance for any larger system that may evolve in future.

As a first step towards developing the system, we collected approximately 140 samples of contemporary ordinary Sanskrit prose from Sanskrit magazines, *pañcatantra* stories and other resources. After manual analysis of this corpus, the reverse rules were created keeping in mind the forward applying rules of Pāṇini. The sections on *subanta* (nominal inflected word) from Pāṇini and later grammarians were studied thoroughly and the following processes were formalized -

Stems were categorized according to their-

1) POS category
2) final sound
3) Gender

The same categorization has been used for generation as well as analysis. There is a set of 21 case affixes which apply to nominal bases under conditions of 8 cases (Nominative, Accusative, Instrumental, Dative, Ablative, Genitive, Locative and Vocative of which the 6th and the 8th are not real cases in Sanskrit) and 3 numbers (singular, dual and plural). The stemming process involves processing the string from right or left looking for important cues based on the last character, POS category and gender/number information. The rule base (created to identify inflection patterns within the string) consists of the abstract forms of all the rules pertaining to these processes from the Pāṇini's grammar. Besides the rulebase, an example-base of most common occurrences was also created. We have listed all suffixes and analysis rules in the chapter four.

Chapter one discusses available morphological analyzers for Indian and Non-Indian languages, current status of R&D in this field, structure and organization of *Aṣṭādhyāyī* (AD), and Nominal Inflectional Morphology (*subanta*) of Pāṇini.

Chapter two discusses Sanskrit sentence and brief description of morphological system of Pāṇini.

Chapter three discusses morphological generation rule and process of Sanskrit nominal inflections and *avyaya*, punctuation, verb and nominal inflectional morphology recognition patterns through machine from the Sanskrit texts.

Chapter four describes the analysis of rules for Sanskrit nominal inflected forms. The rules have been developed by us it is stored in database in Unicode UTF-8 Devnagari format

Chapter five describes implementation methods including formalization of the analysis routines?

Chapter six discusses the web based morphological analyzer for Sanskrit– the front end, Java objects, databases, linguistic resources (corpora, rule bases and example bases), how they work and what is the basic requirement of the system and how to apply *sandhi* and *subanta* rules wherever necessary.

Chapter seven discusses the result analysis, limitations, conclusion and future research and development.

INTERNATIONAL ALPHABET FOR SANSKRIT TRANSLITERATION (IAST)

International Alphabet of Sanskrit Transliteration (IAST)				
अ	आ	इ	ई	उ
a	*ā*	*i*	*ī*	*u*
ऊ	ऋ	ॠ	ऌ	ॡ
ū	*ṛ*	*ṝ*	*ḷ*	*ḹ*
ए	ऐ	ओ	औ	अं
e	*ai*	*o*	*au*	*am*
ः				
ḥ				
क्	ख्	ग्	घ्	ङ्
k	*kh*	*g*	*gh*	*ṅ*
च्	छ्	ज्	झ्	ञ्
c	*C*	*j*	*jh*	*ñ*
ट्	ठ्	ड्	ढ्	ण्
ṭ	*ṭh*	*ḍ*	*ḍh*	*ṇ*
त्	थ्	द्	ध्	न्
t	*th*	*d*	*dh*	*n*
प्	फ्	ब्	भ्	म्
p	*ph*	*b*	*bh*	*m*
य्	र्	ल्	व्	
y	*r*	*l*	*v/w*	
स्	श्	ष्	ह्	ळ्
s	*ś*	*ṣ*	*h*	*ḷ*
क्ष्	ज्ञ्	श्र्	ः	
kṣ	*jñ*	*śr*	*ḥ*	

ACRONYMS AND ABBREVIATIONS

Words	Acronym
‘hari’	H
‘jñāna’	J
‘ramā’	R
‘sarvā’	S
‘sarva’	S
‘vāri’	V
‘viśvapā’	V
Aṣṭādhyāyī	AD
Aṣṭādhyāyī	AD
Application Programming Interface	API
British National Corpus	BNC
Central Electronics Engineering Research Institute	CEERI
Centre for Development of Advanced Computing	CDAC
Communications and Information Technology	MCIT
Compound Derived	CD
Constituent Likelihood Automatic Word-tagging System	CLAWS
dhātupāṭha	DP
Doctor of Philosophy	Ph.D.
Example Base	EB
gaṇapāṭha	GP
Indian Institute of Science,	IISc
Indian Institute of Technology	IIT
Indian Languages Transliteration	ITRANS
Indian Statistical Institutes	ISI
Java Database Connectivity	JDBC
Jawaharlal Nehru University	JNU
Machine Translation	MT
Master of Philosophy	M. Phil.
Microsoft India Ltd.	MSI
National Centre for Software Technology	NCST

Natural Language Processing	NLP
Natural Languages Processing	NLP
Nominal Base	NP
Noun Phrase	NPs
Online Multilingual Amarakoṣa	OMA
Optical character Reader	OCR
pratyāhāra sūtra	PS
Research and Development	R&D
Resource Centre for Indian Languages Technology Solutions	RCILTS
Rule Base	RB
Sūtrapāṭha	SP
Siddhāntakaumudī	SK
Special Centre for Sanskrit Studies	SCSS
University Centre for Computer Corpus Research on Language	UCREL
University Grants Commission	UGC
Verb Phrase	VPs
American and French Research on the Treasury of the French Language	ARTFL

CHAPTER ONE

MORPHOLOGICAL ANALYZERS AND THE SYSTEM OF PĀṆINI

Morphological Analyzers

Morphological analyzers are critical for any useful natural language the system. This is more so for morphology rich Indian languages. Even for configurational languages like English, a syntactic parser will have to go down further to the morphology level. The variation and arbitrariness in the behavior of morphemes have been challenges for most analyzers. Traditionally morphological processors have been used for the following tasks–

- Analysis
  - Taggers
  - Chunkers
  - Word breakers/lemmatizers
- Word building
  - Paradigm generator

Recent advancements in this field for building useful analyzers have been in last 15-20 years. The field has become more applied now than being just concerned with academic research.

1. Approaches and Technology

Computational morphology has been following rule based string processing techniques as well as finite state and statistical techniques depending on the nature of morphology in each language.

1.1 Cut and Paste Method

Cut and paste has been a very popular method in computational morphology. The canonical form is derived by following specified

replacement procedures at the end of a string. The best known ancestor of these systems dates back to the 1960s (Allen, J., Hunnicutt, M. S., and Klatt, D. 1987)[1].

Another system known as MORPHOGEN. Is is a commercial toolkit for creating sophisticated cut and paste analyzers (Petheroudakis, 1991)[2]. Other system called MAGIC is also a cut and paste rule based system in which rules are applied in advance to produce the right allomorph for every allowed combination of a morpheme (Schuller, Zierl, 1993)[3].

1.2 Finite State Techniques

Finite state techniques are used in cases where large lexicons are to be checked. It also explains morphotactics better than the cut-paste method. Automatic recognition and generation of word forms was introduced in early 80s. Rules of morphological alternations could be implemented using FSTs as a finite state network (Johnson 1972, Kaplan and Kay 1994)[4]. First practical application of model appeared in the 90s (Koskenniemi 1983[5], Karttunen 1993[6], Antworth 1990[7], Karttunen and Beesley 1992[8], Ritchie, Russell et al, 1992[9], sproat 1992[10]). These systems

---

[1] Allen, J., Hunnicutt, M. S., and Klatt, D. (1987). *From text to speech—the MITalk system*. MIT Press, Cambridge, Massachusetts.

[2] Petheroudakis, J. (1991). *MORPHOGEN automatic generator of morphological information for base form reduction*. Technical report, Executive Communication Systems ECS, Provo, Utah.

[3] Schuller, G., Zierl, M., and Hausser, R. (1993). MAGIC. *A tutorial in computational morphology*. Technical report, Friedrich-Alexander Universitat, Erlangen, Germany.

[4] Kaplan, Ronald M., and Kay, Martin (1981). *Phonological Rules and Finite-State Transducers*. Paper presented at the Annual Meeting of the Linguistic Society of America. New York.

[5] Kimmo Koskenniemi. *Two-level morphology: A general computational model for word form recognition and production*. Publication No: 11, Department of General Linguistics, University of Helsinki, 1983.

[6] Karttunen, Lauri (1993).Finite-State LexiconCompiler Technical Report. ISTL-NLTT-1993-04-02. Xerox Palo Alto ResearchCenter. Palo Alto, California.

[7] Antworth, Evan L. 1990. PC-KIMMO: a two-level processor for morphological analysis. Occasional Publications in Academic Computing No. 16. Dallas, TX: Summer Institute of Linguistics.

[8] Karttunen and Beesley 1992, A Short History of Two-Level Morphology, Obtained through the Internet: *http://www.ling.helsinki.fi/~koskenni/esslli-2001-karttunen/helsinki.html* (accessed on 12/07/2006).

[9] Ritchie, Russell, et al (1992), Computational Morphology Practical Mechanisms for the English Lexicon, The Mit Press, 1992.

used linked letter trees for the lexicon and parallel FSTs encoding morphemic alternations. The FS techniques are generally used for searching large scale spellchecking wordlists. They also allow bi-directional processing (i.e. both generation and analysis can be performed)

Morphological Analyzers for Non-Indian Languages

Some morph-analyzers have been developed of foreign languages also. Many of these are available on web with details as follows-

1. PC KIMMO

PC-KIMMO is a new implementation for microcomputers of a program called KIMMO after its inventor Kimmo Koskenniemi (Koskenniemi 1983[11]). PC-KIMMO is useful to computational linguists, descriptive linguists, and those developing natural language processing systems. The program is designed to generate (produce) and/or recognize (parse) words using a two-level model of word structure in which a word is represented as a correspondence between its lexical level form and its surface level form. A PC-KIMMO[12] description of a language consists of two files provided by the user:

- a **rules file**, which specifies the alphabet and the phonological (or spelling) rules, and
- a **lexicon file**, which lists lexical items (words and morphemes) and their glosses, and encodes morph tactic constraints.

The two functional components of PC-KIMMO are the generator and the recognizer. The generator accepts as input a lexical form, applies the phonological rules, and returns the corresponding surface form without having to use a lexicon. The recognizer accepts as input a surface form, applies the phonological rules, consults the lexicon, and returns the

---

[10] Richard Sproat (1992), Morphology and Computation, Cambridge, MA, MIT Press.

[11] Kimmo Koskenniemi. *Two-level morphology: A general computational model for word form recognition and production.* Publication No: 11, Department of General Linguistics, University of Helsinki, 1983.

[12] PC-KIMMO: A Two-level Processor for Morphological Analysis (2006), Obtained through the Internet: http://www.sil.org/pckimmo/about_pc-kimmo.html (accessed on 27/04/2006).

corresponding lexical form with its gloss. Figure 1.1 shows the main components of the PC-KIMMO system[13].

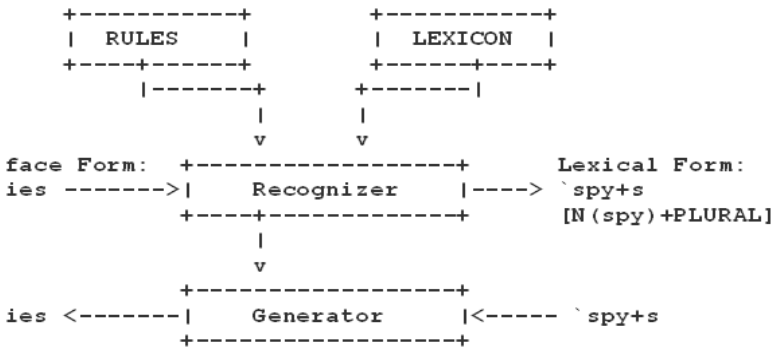

Figure 1.1 Main components of PC-KIMMO

The PC-KIMMO release contains the executable PC-KIMMO program, the function library, and examples of PC-KIMMO descriptions for various languages, including English, Finnish, Japanese, Hebrew, Kasem, Tagalog, and Turkish. These are not comprehensive linguistic descriptions; rather they cover only a selected set of data.

2. CLAWS

CLAWS (Constituent Likelihood Automatic Word-tagging System)[14] POS tagging software for English text has been continuously developed by University Centre for Computer Corpus Research on Language (UCREL) in early 1980s. The latest version of the tagger, CLAWS-4, was used to POS tag 100 million words of the British National Corpus (BNC). CLAWS, has consistently achieved 96-97% accuracy (the precise degree of accuracy varying according to the type of text). Judged in terms of major categories, the system has an error-rate of only 1.5%, with c.3.3% ambiguities unresolved, within the BNC. More detailed analysis of the

[13] PC-KIMMO: A Two-level Processor for Morphological Analysis (2006), Obtained through the Internet: http://www.sil.org/pckimmo/ (accessed on 27/04/2006).

[14] http:// www.comp.lancs.ac.uk/ucrel/claws access on 15th January 2006

error rates for the C5 tag-set in the BNC can be found within the BNC Manual (Garside, R 1987)[15].

3. Arabic Morphological Analysis and Generation

Kenneth R. Beesley at the Xerox Research centre Europe, chemin de Maupertuis, 38240 MEYLAN, France has developed an Arabic Morphological Analysis and Generation using Xerox Finite-State Technology. The system accepts Modern Standard Arabic words and returns morphological analysis and English glosses. Arabic words are displayed in Arabic script using Java applets. This lexicography and Arabic-language consultation for the research system was provided by Tim Buckwalter. The Arabic 'Yarb' font used in the interfaces was created by Yannis Haralambous. System design, interfaces, Arabic-script rendering, and computational linguistics were done by Ken Beesley. Arabic language consulting and lexicography for the commercialization in 2002 was provided by Martine Petrod[16].

4. Comprehensive Morphological Analysis of Chinese, Japanese and Korean Text

Asian language analyzers are used in some of the world's most transaction-heavy environments, like Google's search engine and Amazon's e-commerce site. Rosette Base Linguistics for Chinese, Japanese and Korean are extremely accurate and reliable solutions to help complex applications process unstructured Asian language text by conquering some of these languages' many challenges, such as the use of numerous scripts and absence of spaces between words. Using advanced morphological analysis, Asian Base Linguistics performs functions critical for analyzing Asian text such as segmentation, lemmatization, noun de-compounding,

[15] Garside, R. (1987). The CLAWS Word-tagging System. In: R. Garside, G. Leech and G. Sampson (eds), The Computational Analysis of English: A Corpus-based Approach. London: Longman

[16] Arabic Morphological Analysis and Generation (2005), Obtained through the Internet:
http://www.xrce.xerox.com/competencies/content-analysis/arabic/ (accessed on 11/12/2005)

part-of-speech tagging, sentence boundary detection, and base noun phrase analysis[17].

5. ARIES Natural Language Tools for Spanish

The ARIES Natural Language Tools make up a lexical platform for the Spanish language developed by Natural Language Processing Group (UPM[18]-UAM[19]). These tools can be integrated into NLP applications. They include: a large Spanish lexicon, lexicon maintenance, access tools and morphological analyzer and generator. Non-exclusive, non-transferable licenses are available for the following components[20]:

A. The Prolog GRAMPAL Analyzer/Generator

It is a public domain demonstration system written in Prolog for morphological treatment and lexicon. It includes a small demo lexicon, a DCG grammar for word formation and some predicates to test both analysis and generation. It runs under Sicstus Prolog 2.1.9[21].

B. The Prolog GRAMPAL Dictionary

A collection of Prolog predicates suitable for use with the public domain GRAMPAL DCG grammar. It is capable of generating/ recognizing well-formed inflected forms for verbs, nouns and adjectives. It has no adverbs, determiners, conjunction, prepositions, etc. It does not treat clitic pronoun attachment or derivatives[22].

---

[17] Obtained through the Internet: *http://www.basistech.com/base-linguistics/asian/* (accessed on 11/12/2008)

[18] UPM (2010) Obtained through the Internet: *http://www.dit.upm.es/* (accessed on 11/10/2010)

[19] UAM (2010), Obtained through the Internet: *http://lola.lllf.uam.es/* (accessed on 01/11/2010)

[20] Jose C.Gonzalez, Jose M. Goni, Amalio Nieto, Antonio Moreno, and Carlos A. Iglesias. The ARIES Toolbox: a continuing R+D Effort. In David Farwell, editor, Proceedings of the International Workshop on Spanish Language Processing Technologies, SNLP-97, Santa Fe, New Mexico, USA, July 1997

[21] The Prolog GRAMPAL analyser/generator (2005), Obtained through the Internet: http://www.mat.upm.es/~aries/description.html (accessed on 27/04/2005).

[22] The Prolog GRAMPAL Dictionary (2005), Obtained through the Internet: http://www.mat.upm.es/~aries/description.html (accessed on 27/12/2005).

C. The expanded ARIES Dictionary

It is a collection of expanded entries (allomorphs) with morphological information and contains a full set of morphemes dealing with clitic pronoun attachment (but without verb marking for correct attachments). It includes information about some derivative morphological processes (inflected adjectives from past participles and adverbs ended in '*mente*' from adjectives)[23].

D. The Source ARIES Lexical Base

It is a collection of inflectional models, rules for off-line computing of allomorphs, unexpanded lemma entries, lexicalized irregular words. It is the most complete source of information we have available and the most useful for dictionary maintenance. A tool for expanding the source dictionary to the expanded dictionary is also provided. The current size of this lexicon is 38,500 lemma entries (21,000 nouns, 10,000 adjectives, 7,500 verbs and 500 auxiliary words) plus more than 600 inflectional morphemes[24].

E. Access Tools

The C/C++ programming interface for lexical access to the ARIES dictionary: It is a set of tools and libraries to build tree indexes to the allomorph dictionary and to retrieve them by an application[25].

F. Morphological Analyzer

The C/C++ morphological analyzer that makes use of the lexical interface mentioned above. This permits to improve efficiency by integrating word segmentation with lexical access also. By now, it is a (pseudo)-unification chart based parser for context-free morphological grammars.

These system supported platforms are UNIX and DOS Operating System with GNU gcc/g++ (djgpp for DOS) compilers. Tools have been tested MSDOS, HP-UX 9.05, SunOs 4.1.3 and Solaris 2.4[26].

---

[23] The expanded ARIES Dictionary (2005), Obtained through the Internet: http://www.mat.upm.es/~aries/description.html (accessed on 27/12/2005).

[24] The Source ARIES Lexical Base (2005), Obtained through the Internet: http://www.mat.upm.es/~aries/description.html (accessed on 27/11/2005).

[25] Access Tools (2006), Obtained through the Internet: http://www.mat.upm.es/~aries/description.html (accessed on 27/11/2006).

[26] ARIES Natural Language Tools (2006), Obtained through the Internet: www.mat.upm.es/~aries/description.html (accessed on 27/05/2006)

6. Morphological Analysis of Bulgarian Sentence

For a couple of years a system of machine dictionaries a lexical database for the Bulgarian language and a morphological processor has been developed at the Department of Computer Science of the University of Plovdiv. The system for lexical database management is situated on a WEB server[27].

7. French Morphological Analyzer

The American and French Research on the Treasury of the French Language (ARTFL) Project: morphological analysis using the INFL analyzer allows you to enter one or more French words (lower case only, no punctuation) at the prompt and returns the context-free morphological analysis for each[28].

8. Greek Morphological Analyzer

The Perseus Project: morphological analysis using the morphological analyzer allows you to enter one or more Greek words in Latin transliteration at the prompt and returns the morphological analysis for each term[29].

9. Latin Parser and Translator 0.96

This is a Visual Basic program which translating from Latin into English developed by Adam McLean. The alternative translations for ambiguous words have been extended, and the user can now edit, within the program, the Latin as well as English translation files. Version 0.96 has fixed a number of minor bugs in the Latin parsing, added some extra help information, changed the program layout a little, and now comes with an installer[30].

---

[27] Bulgarian language and a morphological processor (2006), Obtained through the Internet: http://www.uni-plovdiv.bg/dcs/morphe.htm (accessed on 20/04/2006)

[28] The ARTFL Project (2006), Obtained through the Internet: http://humanities.uchicago.edu/orgs/ARTFL/forms_unrest/analyze.query.html (accessed on 20/03/2006)

[29] Greek Morphological Analyzer (2006), Obtained through the Internet: http://www.perseus.tufts.edu/cgi-bin/morphindex (accessed on 07/05/2006)

[30] Latin parser and translator (2006), Obtained through the Internet: http://www.alchemywebsite.com/latin/latintrans.html (accessed on 07/05/2006)

10. Multilingual Verb Conjugator

Multilingual verb conjugator, conjugate at least some of the regular verbs in 27 different languages. Database of this system have been cover 27 different languages including Italian, French, Spanish, Polish, German, Esperanto, English, Latin, Portuguese, Greek, Finnish, Czech, Croatian, Sicilian. It is a work in progress that still lacks many verbs[31].

11. Turkish Morphological Analyzer

This analyzer has been developed using the two-level transducer technology developed by Xerox. It can process near about 900 forms per second. This implementation of Turkish uses about 30,000 Turkish root words[32].

Morphological Analyzers for Indian Languages

The R & D for morphological analyzers for Indian languages was spearheaded by setting up of RCILTS (Resource Center for Indian Languages Technology Solutions) by the Ministry of Communications and Information Technology (MCIT)[33]. The leading resource centers are –

- IIIT, Hyderabad
- IIT Kanpur
- IIT Kharagpur
- IIT Mumbai

Much work in the area of NLP in India has been carried out is still on at several places and in several languages. Of particular mention is the works carried out in

- National Centre for Software Technology (NCST)
- Indian Statistical Institutes (ISI)
- Thapar Institute of Engineering and Technology

[31] Multilingual verb conjugator (2006), Obtained through the Internet: http://www.nlp.cs.bilkent.edu.tr/cgi-bin/tma (accessed on 25/01/2006)
[32] Turkish Morphological Analyzer (2006), Obtained through the Internet: http://www.semarweb.com/morphology.html (accessed on 22/02/2006)
[33] MCIT (2007), Obtained through the Internet: http://www.mcit.gov.eg/ (accessed on 22/02/2007)

- Utkal University, Anna University
- Chennai, Bhubaneshwar
- A tagged text corpus developed from using the web as source of data in Bengali at Jadavpur University,
- Kolkata and University of Hyderabad,
- Indian Institute of Science, (IISc) Bangalore
- Central Electronics Engineering Research Institute (CEERI), Pilani
- Tata Institute of Fundamental Research, Mumbai
- IBM, India research lab, Microsoft India, Tata Consultancy Services, HP, HCL and Webdunia etc.
- CDAC, Kolkata

Work on Sanskrit informatics has been going on at the following places

- C-DAC (Pune), CDAC Hyderabad and Bangalore
- Special Centre for Sanskrit Studies, Jawaharlal Nehru University, New Delhi
- Vanashtali Vidyapeeth, Rajasthan
- Rastriya Sanskrit Vidyapeeth Tirupathi
- Lal Bahadur Shastri Rastriya Sanskrit Vidyapeeth, New Delhi.
- Academy of Sanskrit Research, Melkote, Mysore[34]

1. Morphological Analyzers by Akshara Bharathi Group

Morphological analyzer for Sanskrit, Telugu, Hindi, Marathi, Kannada and Punjabi have been developed by Akshara Bharathi group at Indian Institute of Technology, Kanpur, India and University of Hyderabad, Hyderabad, India (funded by Ministry of Information Technology, India) and claim for the 95% coverage for Telugu (for arbitrary text in modern standard Telugu) and 88% coverage for Hindi.

[34] Academy of Sanskrit Research, Melkote, Mysore (2007), Obtained through the Internet: http://www.sanskritacademy.org/Research.htm (accessed on 22/03/2007)

2. Hindi-Marathi-Telugu Morphological Analyzers

This morphological analyzer[35] allows you to choose your language and your font. It includes links to pages with other linguistic resources for Indian languages and English, dictionaries the National Institute of Information Technology in India.

3. Assamese and Manipuri Morphological Analyzers

The Resource Centre for Indian Languages Technology solution Indian Institute Technology Guwahati has developed two morphological analyzers for Assamese and Manipuri. They have been used both in the spell checker and OCR systems. Both the morphological analyzers use the technique of stemming where in the affixes are either deleted or added to arrive at the root words[36].

Sanskrit Morphology

The study of the structure and form of words in languages or a language, including inflection, derivation, and the formation of compounds is called morphology. In Sanskrit, a syntactic unit is called *pada.* Cordona[37] (1988) posits the formula for Sanskrit sentence **(N-En) p…(V-Ev)p**. *Pada* can be nominal (*subanta*) or verbal (*tiṅanta*). These forms are formed by inflecting the stems and hence they are part of Sanskrit inflectional morphology. The derivational morphology in Sanskrit studies primary forms (*kṛdanta*) and secondary forms (*taddhitānta*), compounds (*samāsa*), feminine forms (*strī pratyayānta*) etc.

Sanskrit has two types of morphology- nominal (consisting of primary and secondary derivations of nouns including feminine forms and compounds and their inflected forms called *padas*) and verbal (consisting primarily of inflected forms and also of derived verbs). Sanskrit has approximately 2014 verb roots (including *kandvādi*), classified in 10

[35] Hindi-Marathi-Telugu Morphological Analyzers (2007), Obtained through the Internet: http://www.iiit.net/ltrc/morph/morph_analyser.html (accessed on 22/02/2007)

[36] Assamese and Manipuri Morphological Analyzers (2008), Obtained through the Internet: *http://www.iitg.ernet.in/rcilts/morph.html* (accessed on 22/02/2008)

[37] George Cardona, 1988 Pãnini, His Work and its Traditions, vol ... i (Delhi: MLBD, 1988)

groups (*gaṇas*), the derived verb forms can have 12 derivational suffixes[38]. These can have *ātmanepadi* and *prasmaipadi*. A verb root may have approximately 2190 (tense, aspect, number etc.) morphological forms. Sanskrit Nominal inflectional morphology is two types, Primary [*kṛdanta* (roots forms that end with *kṛt* suffixes)] and secondary [*taddhitānta* (noun forms that end with *taddhita* suffixes)]. Secondary nominal inflectional morphology may be of following types like- *samāsānta* (compound nouns), *strīpratyayānta* (feminine forms) etc. They can also include *upasargas* (prefix) and *avyayas* (indeclinables) etc. According to Pāṇini, there are 21 morphological suffixes (seven *vibhaktis* and combination of three numbers = 21)[39] which are attached to the nominal bases (*prātipadika*)[40] according to syntactic category, gender and end character of the base[41].

The System of Pāṇini and Nominal Inflectional Morphology

1. The System of Pāṇini

Pāṇini's grammar Aṣṭādhyāyī [(AD) (approximately 7th century BCE)] is important for linguistic computation for two reasons. One, it provides a comprehensive and rule based account of a natural language in about 4000 rules - the only complete grammatical account of any language so far. Two, the model of a 'grammar-in-motion' that it provides seems to closely mimic a fully functional Natural Language Processing (NLP) system[42]. Aṣṭādhyāyī (7th BCE) is a composite text including the following modules –

1. *śivasūtra* or *pratyāhārasūtra* (14) (PS)

---

[38] सन्-क्यच्-काम्यच्-क्यङ्-क्यषोथाचारक्विब्-णिज्यङ्स्तथा | यगाय ईयङ् णिङ् चेति द्वादशमी सनादयः || *[IAST: san-kyac-kāmyac-kyaṅ-kyaṣothācārakvib-ṇijyaṅstathā. yagāya īyaṅ ṇiṅ ceti dvādaśamī sanādayaḥ]*

[39] स्वौजसमौट्छष्टाभ्याम्भिस्ङेभ्याम्भ्यस्ङसिभ्याम्भ्यस्ङसोसांङ्योस्सुप्-4.1.2 *[IAST: svaujasamauṭchṣṭābhyāmbhisṅebhyāmbhyasṅasibhyāmbhyasṅsosāṃṅyossup]*

[40] अर्थवद्धातुरप्रत्ययः प्रातिपदिकम्-1.2.45, कृत्तद्धितसमासाश्च-1.2.46] *[IAST: arthavadhāturapratyayaḥ prātipadikam, kṛttaddhitasamāsāśca]*

[41] Mishra Sudhir K, Jha Girish Nath, 2005, Identifying verb inflections in Sanskrit morphology Proc. of SIMPLE05, IIT Kharagpur.

[42] Jha Girish Nath, 'The System of Panini' Language in India, volume 4:2 February 2004.

2. *śabdānuśāsana* or *sūtrapāṭha* (3965 or 3983 in *kāśikāvṛtti* (SP)
3. *dhātupāṭha* (1967 verb roots - 2014 including *kaṇḍvādi* roots) (DP)
4. *gaṇapāṭha* (other pertinent items like primitive nominal bases, *avyayas*) (GP)

Sūtrapāṭha (SP) is arranged in eight *adhyāyas* (chapters) each divided into four sub-chapters (*pādas*). The SP has approximately 3965 rules (*sūtras*) which have been arranged in X.X.X format (to be accessed in as *adhyāya.pāda.sūtra* format)[43]

The following is a summary of topics discussed in the Aṣṭādhyāyī[44] -

Chapter I

Major definitional and interpretational rules
Rules dealing with extension (*atideśa*)
Rules dealing with *atmanepada-parasmaipada*
Rules dealing with the *kāraka*

Chapter II

Rules dealing with compounds (*samāsa*)
Rules dealing with nominal inflection
Rules dealing with number and gender of compounds
Rules dealing with replacements relative to roots
Rules dealing with delection by *luk*

Chapter III

Rules dealing with derivational of roots ending in affixes *san* etc.
Rules dealing with the derivational of ending in a *kṛt*
Rules dealing with the derivational of ending in a *tiṅ*

Chapter IV

Rules dealing with derivation of a *pada* ending in a *sup*
Rules dealing with feminine affixes
Rules dealing with the derivational of nominal stems ending in an affix termed *taddhita*

Chapter V, VI & VII

[43] Jha Girish Nath 'The System of Panini' Language in India, volume 4:2 February 2004
[44] Sharma, Rama Nath, The Aṣṭādhyāyī of Pāṇini – Volume-I page-75-76

Rules dealing with doubling
Rules dealing with *samprasāraṇa*
Rules dealing with the *saṁhitā*
Rules dealing with the augment (*āgama*) *suṭ*
Rules dealing with accents
Rules dealing with phonological operations relatives to a pre-suffix base (*aṅga*)
Rules dealing with operations relative to affixes augment etc.

Chapter VIII
Rules dealing with doubling (*dvitva*) relative to a *pada*
Rules dealing with accent relative to a *pada*
Rules dealing with other phonological relatives to a *pada*
Rules dealing with miscellaneous operations relative to a non-*pada*

Kapoor (1992) has reduced the treatment of subject matter into four divisions[45]: Chapters 1-2 dealing with classification and enumeration of bases and categories, Chapters 3-5 consist of *prakṛti-pratyaya* enumeration, and derivation of bases, Chapters 6-8.1 deal with the synthesis of *prakṛti-pratyaya*, and Chapters 8.2-8.4 deal with the rules of morphophonemic.

1.1 The Modules of Aṣṭādhyāyī (AD)

The AD divided into eight chapters. There is lot of modules. Here we are giving detail only basic modules. The basic modules are-

1.1.1 śiva sūtras or pratyāhāra sūtra (PS)

The purpose of the PS component is to give a list of all Sanskrit phonemes. But rather than listing just the phonemes, the *sūtras* in the PS component are interspersed by meta-linguistic markers, called anubandhas. By a well-defined method *(ādirantyena sahetā*), Pāṇini creates variables or macros to be used in his grammar. For example, '*al*' refers to the list of all phonemes; '*ac*' refers to all vowels, '*hal*' to all consonants and *ñam* to all nasals. The 14 *sūtras* are-

The first 4 *sūtras* cover all the vowels and the last 10 *sūtras* include all the consonants. Again, all vowels and consonants of Sanskrit have been arranged in such a way in these *sūtras* that they can be referred to without mentioning them separately.

[45] Kapoor, Kapil, *Text Interpretation: The Indian Tradition*

Of the hundreds *pratyāhāras* that could in principle be formed from these *sūtras*, Pāṅini introduced only 43 *pratyāhāras* in Aṣṭādhyāyī. Another *rañ=(r, l)*][46] *pratyāhāra* introduced by later grammarians. Some *pratyāhāra* are ambiguous, for example, '*ṇ*' occurs twice in the list, which means that you can assign two different meanings to *prtyāhāras aṇ* (including or excluding *ṛ* etc.)[47]

Table 1-1. *śiva sūtras* or *pratyāhāra sūtra* (PS)

Sr.	pratyāhāra sūtra	
	Devanagri	**IAST**
1	A C E hÉ	*a i u ṇ*
2	G I MÜ	*ṛ ḷ k*
3	L AÉ XÌ	*e o ṅ*
4	LÌ AÉ cÉ	*ai au c*
5	WÌ rÉ uÉ U OÌ	*h y v r ṭ*
6	sÉ hÉ	*l ṇ*
7	gÉ qÉ XÌ hÉ IÉ qÉ	*ñ m ṅ ṇ n m*
8	fÉ pÉ gÉ	*jh bh ñ*
9	bÉ RÌ kÉ wÉ	*gh ḍh dh ṣ*
10	eÉ oÉ aÉ QÌ S vÉ	*j b g ḍ d ś*
11	ZÉ T üNÌ P ¡jÉ cÉ O iÉ uÉ	*kh ph ch ṭh th c ṭ t v*
12	MÜ mÉ rÉ	*k p y*
13	vÉ wÉ xÉ UÌ	*ś ṣ s r*
14	W sÉ	*h l*

[46] ङणटञ्वात् स्मृतो हृयेकः, चत्वारश्च चमान्मताः |
[*IAST: ṅaṇaṭañvāt smṛto hṛyekaḥ, catvāraśca camānmatāḥ*], शलाभ्यां षड् यरात्पञ्च, षाद् द्वौ च कणतस्त्रयः || *[IAST: śalābhyāṃ ṣaḍa yarātpañca, ṣāda dvau ca kaṇatastrayaḥ]*, केषाञ्चिच्च मते रोऽपि, प्रत्याहारोऽपरो मतः | *[IAST: keṣāñcicca mate ropi, pratyāhāro'paro mataḥ]* लस्थाऽवर्णेन वाञ्छन्त्यनुनासिकबलादिः ||
[IAST: lasthā'varṇena vāñchantyanunāsikabalādiḥ]

[47] अणुदित्सवर्णस्य चाप्रत्ययः *[IAST: aṇuditsavarṇasya cāpratyayaḥ]*

1.1.2 sūtrapāṭha (SP)

The SP contains about 3965 sūtras or 3983 in *kāśikāvṛtti* arranged in chapters (*adhyāya*) and sub-chapters (*pāda*) in a particular order[48].

Table 1-2. Distribution of AD *sūtras*

Chapter	Pāda I	Pāda II	Pāda III	Pāda IV	Total Rules
1st	74	73	93	109	349
2nd	71	38	73	85	267
3rd	150	188	176	117	631
4th	176	144	166	144	630
5th	135	140	119	160	554
6th	217	198	138	175	728
7th	103	118	119	97	437
8th	74	108	119	68	369
Total Rules in Aṣṭādhyāyī			**3965**		

sūtras are verb-less sentences unlike those in natural language and give an impression of formula or program like code. They are of following types[49] –

1. Samjña (Technical Rules): Rules which assign a particular term to a given entity.
2. Paribhāṣā (Interpretive Rules): Rules which regulate proper interpretation of a given rule or its application.
3. Vidhi (Operational Rules): Rules which state a given operation to be performed on a given input.
4. Niyama (Restriction Rules): Rules which restrict the scope of a given rules.
5. Atideśa (Extensions Rules): Rules which expand the scope of a given rules, usually by allowing the transfer of certain properties which were otherwise not available.
6. Adhikāra (Heading Rules): Rules which introduce a domain of rules sharing a common topic, operation, input, physical arrangement, etc.

---

[48] Shastri, Bheemsen, Laghusiddaantkaumudi Ist part, page: 5
[49] संज्ञा च परिभाषा च विधिर्नियम एव च। अतिदेशोऽधिकारश्च षड्विधं सूत्रमुच्यते॥
[IAST: saṃjñā ca paribhāṣā ca vidhirniyama eva ca. atideśoऽ dhikāraśca ṣaḍvidhaṃ sūtramuccyate..]