

# Rule based POS Tagger for Sanskrit

Sharada Adinarayanan, J. Naren, P. Sriranjani and Dr.G. Vithya

**Abstract---** POS tagging is a process of attaching each word in a sentence with a suitable tag from the given set of tags. In the paper, rule based view of NLP is taken up for tagging the part of speech for Sanskrit words. The foundation for POS tagging is morphological analysis. The twelfth chapter of Bhagavad Gita is considered as input for POS tagging process. Annotated corpora will be developed and used for retrieving the grammatical category of the input text. Sanskrit is a language with very concrete grammar proposed by Panini (4000.B.C) and has layered grammatical structure. Thus rule based approach would fulfill the tagging process rather than stochastic or probabilistic approach (existing system). Therefore, the project aims to improve the accuracy by utilizing the efficient lookup strategies, searching and sorting techniques and finally rule formations (utilizing the richness of Sanskrit grammar) to quickly narrow down the assignment of grammatical category to words. The major challenge is the tokenization process of joined words. Since Sanskrit has many inflected noun and verb forms, identifying the correct grammatical category involves contextual meaning and semantics to be taken into view. Also, semantic analysis, derivative analysis and Sandhi analysis is done.

**Keywords---** Annotated Corpora, Tokenization, Morphological Analysis.

---

## I. INTRODUCTION

Parts-of-Speech Tagging is a process of assigning appropriate tags to each word in a given unprocessed sentence. Parts -of -speech for a language according to the grammar can be categorized as noun, verb, adverb, pronoun, adjectives, conjunctions and their remaining categories. POS tagger performs all operations by referring dictionaries, rules and tag set.

The ambiguity is resolved by using probabilistic method. There are three techniques in POS tagging: Rule- based Model, Stochastic Model and Hybrid Model. Rule-based Model is one of the oldest approaches, which uses a set of Hand-coded rules to assign the appropriate tags for the given input. The accuracy remains high if more number of tagged words and hand coded rules are used. The Stochastic Model uses the concept of picking the most appropriate tag word in the unprocessed text. The various techniques used in stochastic model are Hidden Markov Model, Maximum Entropy Markov Model and Conditional Random Field Model. Hybrid model uses the combination of both Rule-based as well as stochastic model wherein the hand-coded rules are combined with techniques present in the Stochastic Model.

---

Sharada Adinarayanan, B.Tech Computer Science and Engineering, School of Computing, SASTRA Deemed University, Thanjavur, Tamil Nadu, India. E-mail: sharadha.adinarayanan@gmail.com

J. Naren, Assistant Professor, School of Computing, SASTRA Deemed University, Thanjavur, Tamil Nadu, India. E-mail: naren.jeeva3@gmail.com

P. Sriranjani, B.Tech Computer Science and Engineering, School of Computing, SASTRA Deemed University, Thanjavur, Tamil Nadu, India. E-mail: nsriranjanie@gmail.com

Dr.G. Vithya, Professor, School of Computing, KL University, Vijayawada, AP, India. E-mail: vithyamtech@gmail.com

## II. LITERATURE SURVEY

The paper by A.J.P.M.P.Jayweera.et.al has proposed Hidden Markov Model based part of speech tagging and lexical semantics for Sinhala Language. Sinhala is the native language of ethnic group of Sri Lanka, spoken by more than seventeen million people. Sinhala is one of the morphologically rich languages of great complexity, and with various kinds of words in inflected form, for which grammatical tagging is very essential. The tag set comprises of 26 tags. Sinhala Text corpus containing 90551 words from 2754 sentences gathered from Sinhala newspapers. Beta version of corpus developed by UCSC is used. Viterbi algorithm is used for finding the best tag sequence based on the text corpus. Input to the algorithm is a string of words and tag set, the output is single best tag for each word. Supervised machine learning approach is used for training the tagger to find the transition probability of the given sequence of words. Statistical based approach is used and an accuracy of about 90.91% for known words is achieved. [1]

The paper by R Muni Prashanthi.et.al, have presented Implementation of Tree Tagger for Sanskrit. Annotation of text with POS and lemma information is done using Tree tagger tool. Tree Tagger involves two phases which are training and testing phases. Three files are used in training phases consisting of open class file, parameter file and lexicon. Two programs namely training and tagger programs are involved in the Tree Tagger. Tag set developed by J.N.U, New Delhi for Sanskrit is used. The training set was annotated using a tag set consisting of 134 tags. Errors were investigated and the reasons were analyzed. Tree tagger for Sanskrit yielded above average results though not better than the performance for English language. [2]

The paper by Pallavi Bagul.et.al, Archana Mishra, deals with POS Tagging of Marathi language text. The word in a given sentence is tagged by using rule based approach. Tokenization is used to split the given string of inputs into tokens, and the result of this process i.e. tokens, are compared with WordNet to assign the correct POS Tags for each token. Ambiguity in Marathi is resolved by using Marathi grammar rules. WordNet, corpus, tagset has been used as a database for providing correct POS tags to every word in a given sentence. Accuracy can be increased by increasing the number of tagged words in WordNet. [3]. The paper by Kanak Mohnot .et.al has presented A Hybrid Based Part of Speech Tagger for Hindi. Corpus of over 80,000 words having 7 standard different part of speech tags is evaluated by the proposed system for Hindi Language. Also accuracy of the proposed tagger is determined. The input words are searched for in the database and are tagged if present, else HMM model is used. The basic dataflow of the proposed system involves:-accepting user text, tokenizing the text, converting them to singular form, applying rules if word is found in database else assign category using Hindi NER then apply rules by predictive analysis else if multiple Tag condition is encountered, then Hindi NER and HMM are used. The average accuracy the system has attained for POS Tagging is 89.9%. [4]

## III. PROBLEM ASSESSMENT

The project deals with using Rule-Based technique of NLP (Natural language Processing) to perform POS (Parts-Of- Speech) Tagging for Sanskrit Text by assigning appropriate tags according to the correct grammatical categories to the words in unprocessed sentences. Unavailability of considerable amount of annotated corpora of sound quality for South-Asian languages like Sanskrit and tokenization of joined words are the major challenges.

#### IV. PROPOSED SYSTEM

The proposed system uses Rule-based model for POS tagging. The objective of the study is to analyze and improve accuracy of POS Tagging process for Sanskrit words in an unprocessed sentence by using Rule-Based technique of NLP wherein hand-coded rules are used for assigning the tags according to Sanskrit Grammar. [5] Since rule-based approach is proposed, if specific rules are unavailable for tagging a specific category of words, tagging process does not take place as compared to the stochastic model (existing system) where probabilistic frequency is taken into account even if rules are unavailable and tags are assigned. The assumed rule may or may not be correct although in most cases it is correct. Rule-based Approach removes the probabilistic nature of stochastic model and assigns rules only if available thus improving the correctness of the tagged words in an unannotated Sanskrit Text. [8][9]

#### V. SYSTEM DESIGN

The design of the system is based on the following Data- Flow diagrams. The proposed work is divided into 4 individual models assisting POST for Sanskrit. The first module is POS Tagger Module, the second is Semantic Analyzer Module, the third is Derivative Analyzer and the Fourth is Sandhi Analyzer. The first illustrates data-flow diagram for POS Tagger.

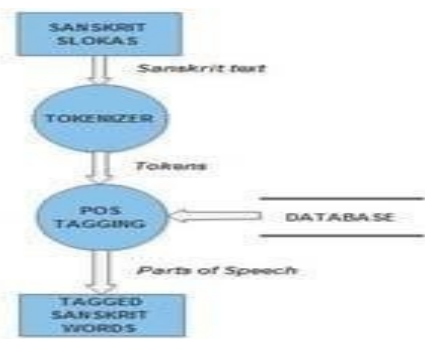


Figure 1.5.1: Data Flow Diagram for POS Tagging

The second illustrates the data-flow diagram for Semantic Analyzer.

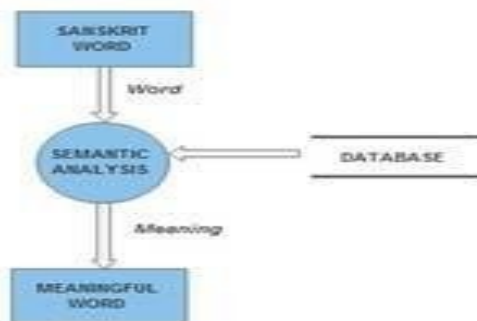


Figure 1.5.2: Data Flow Diagram for Semantic Analysis The third illustrates the data-flow diagram for Derivative Analyzer

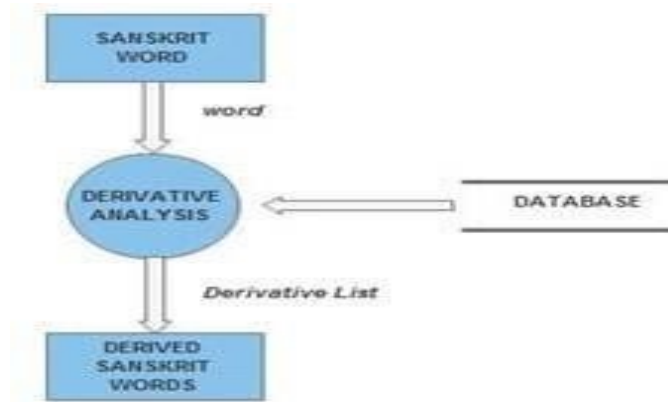


Figure 1.5.3: Data Flow Diagram for Derivative Analysis The fourth diagram illustrates the data-flow diagram for Sandhi Analyzer

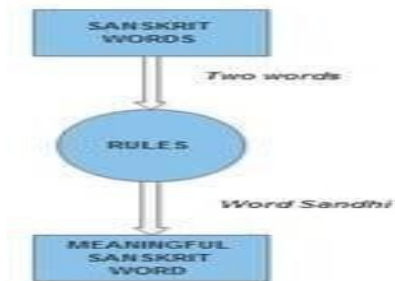


Figure 1.5.4: Data Flow Diagram for Sandhi Generator

## VI. SYSTEM IMPLEMENTATION

A Rule-based technique for POS Tagging for Sanskrit Text is proposed. A specific domain, Sanskrit Slokas from 12<sup>th</sup> chapter of Bhagavad Gita is chosen. Hand-coded rules are used for the tagging purpose with No specific Algorithm is usage. Derivative analysis, Semantic analysis and Sandhi analysis is also done.

### *POS Tagger*

A Rule-Based POS tagger for Sanskrit is proposed. Sanskrit language has concrete grammar rules. Unannotated Sanskrit text is taken as input, tokenized into words and processed to produce tagged Sanskrit words as output. The 12<sup>th</sup> chapter Slokas from Bhagavad Gita is taken as input text for Part-Of-Speech Tagging. The Slokas are categories according to their Sanskrit Grammar and tags are assigned. [1][6][7]

The steps followed for POS Tagging are as follows:

Step1: Stemming – Manual Stemming is done. The stemmed words are populated in database. Step2: Suffix-Stripping-The end string or simply the suffix of each word is stripped. The hand- coded rules are checked with the suffixes.

Step3: Tag Assignment- The appropriate tags are checked for in the database and in the program to assign the correct tags to the words in sentence.

root	Adverb	Prefix	Noun	Verb	Pronouns	Pronouns	Pronoun
ca	pari		satayukta	bhava	ye	aham	sarvaani
api	ati		bhakta	asa	tesaam	mayi	etat
tu	abhi		aksara	kara	yen	maam	idam
niveshaya	prati		avyakta	kru	sah	tvaam	yasmaat
sarvatra	pra		yogavittama	hrusha	yah	rama	maam
evam	upa		mana	shoo	me	rama	rama
hi	adhi		nityayukta	esa	sarva	rama	rama
atah	tu		shradhdhaya	aaveshya	te	rama	rama
ava	para		upeta	tataah	ke	rama	rama
na	pra		yuktatama	aasakta	mad	rama	rama
uurdhvam	vi		mata	avaapyate	rama	rama	rama
atha	ni		anideshya	dhyayanta	rama	rama	rama
samaadhaatun	anu		sarvatra	vas	rama	rama	rama
sthiram	ava		kuutastha	samshayah	rama	rama	rama
tatah	pra		achala	shaknosi	rama	rama	rama
aaptum	ni		dhruva	iccha	rama	rama	rama
sannyasya	upa		endriyagraam	ase	rama	rama	rama
madartham	tu		sambuddhi	kur	rama	rama	rama
karum	adhi		sarvabhootahi	aap	rama	rama	rama
anantaram	ati		kleshah	arpi	rama	rama	rama

Figure 1.6.1: The POS Database

**Semantic Analyzer**

Semantics refers to finding the meanings of words in a sentence. Sanskrit words with their meanings are fed in database table. Words from the Sanskrit Slokas of the 12<sup>th</sup> chapter of Bhagavad Gita are taken as input for which semantic analysis is done. [2]

The steps followed for Semantic Analysis is as follows: Step 1: Get a word as input. Step 2: Check if the word is present in the database. Step 3: Print its meaning

Step 4: If word is not present, check if the root is present in the database.

Step 5: Strip the suffix if root is present in database and perform Semantic analysis

root1	means
dhava	jump
yaacha	beg
bhaja	chant
aas	desire
aap	get
bhava	to be
evam	thus
ye	who
tu	indeed
dharmaamrta	immortal dhar
idam	this
yathayukta	as declared
pariyupaasate	follow
shraddadhaan	endued with s
matparamaah	regarding me e
bhaktaah	devotees
te	they
atilva	exceedingly
me	to me
priyaah	dear
yah	who
na	not
hrsyati	rejoices

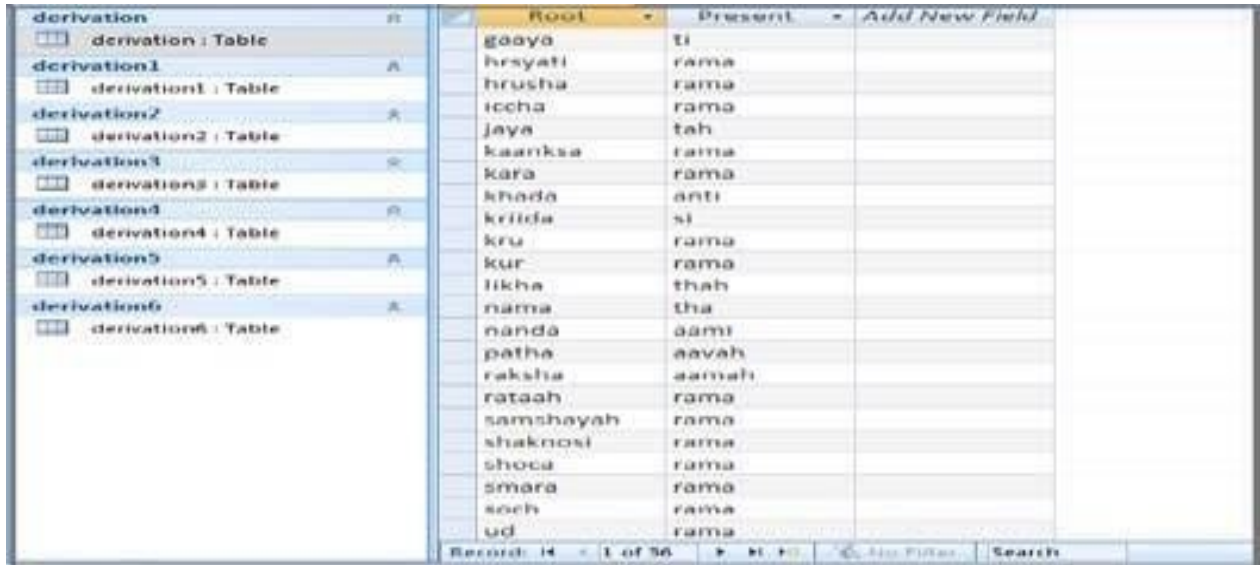
Figure 1.6.2: The Semantic Analyzer Database

### ***Derivative Analyzer***

Derivatives are also known as Kridantas in Sanskrit. The root words and the corresponding derivatives are fed in the database. The derivatives for the parasmaipadi and aatmanepadi forms are displayed as output. [4]

The steps followed for Derivative Analysis is as follows: Step 1: Get a word as input. Step 2: Check if the word is present in database.

Step 3: If the word is present, the derived words of the input root word are displayed.



derivation	Root	Present	Add New Field
derivation1	gaaya	ti	
derivation2	hrsyati	rama	
derivation3	hrusha	rama	
derivation4	iccha	rama	
derivation5	jaya	tah	
derivation6	kaanksa	rama	
derivation7	kara	rama	
derivation8	khada	anti	
derivation9	kriida	si	
derivation10	kru	rama	
derivation11	kur	rama	
derivation12	likha	thah	
derivation13	nama	tha	
derivation14	nanda	aami	
derivation15	patha	aavah	
derivation16	raksha	aamah	
derivation17	rataah	rama	
derivation18	samshayah	rama	
derivation19	shaknosi	rama	
derivation20	shoca	rama	
derivation21	smara	rama	
derivation22	soch	rama	
derivation23	ud	rama	

Figure 1.6.3: The Derivative Database

### ***Sandhi Analyzer***

Sandhis are syllables that help in joining two distinct meaningful words to produce another meaningful word. Sandhis mainly occur in many Indian Languages. Sanskrit consists of complex Sandhi rules. There are three main categories of Sandhi namely: Swara Sandhi, Vyanjana Sandhi and Visaraga Sandhi. The Sandhi module deals with Basic Swara Sandhi Generator. There are 8 types of Swara Sandhi. [3]

The steps followed for Sandhi generator is as follows: Step 1: Get two input strings. Step 2: Apply Basic Swara Sandhi Rules.

Step 3: Display the output string after Sandhi processing.

### ***Graphical User Interface***

Java Jframes are used to design Graphical User Interface which is easy to use and User-friendly. The Graphical User Interface is designed for all four modules. The different parts of the GUI for different modules are shown in the screenshots below.

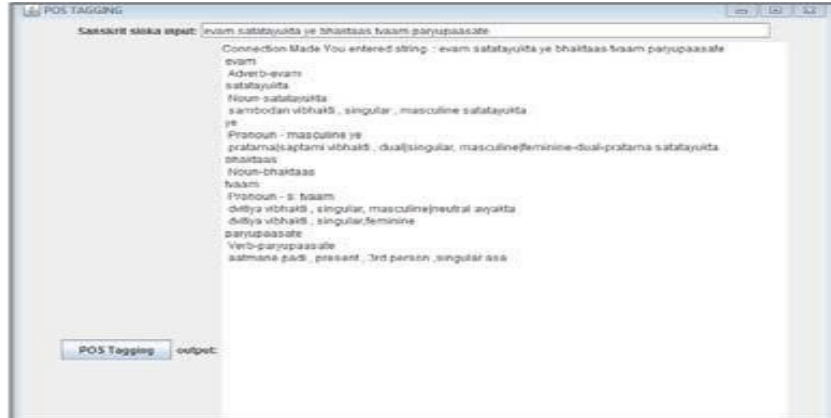


Figure 1.6.5: POS Tagger



Figure 1.6.6: Semantic Analyzer



Figure 1.6.7: Derivative Analyzer

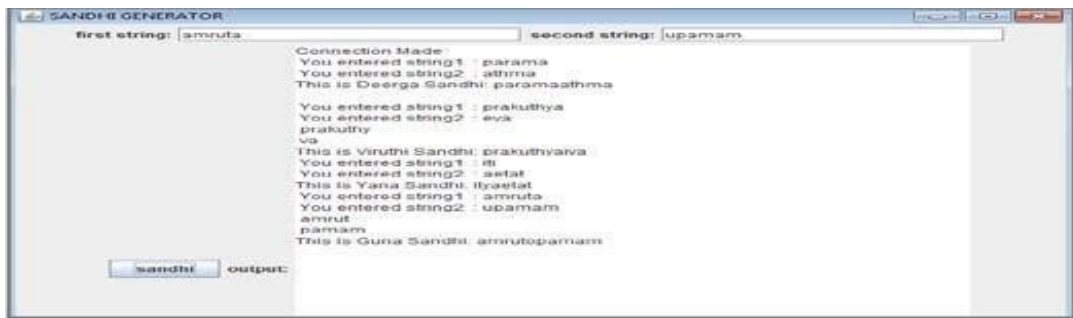


Figure 1.6.8: Sandhi Analyzer

**A. Output for POS Tagger is as given below**

You entered string: evam satatayuktaah ye bhaktaas tvam pary upaasate  
 Evam-Adverb Satatayuktaah-Noun sambodan vibhakti, singular, masculine  
 satatayukta-root Ye-pronoun, masculine  
 Bhaktaas-NounPratama vibhakti, singular, masculine  
 Tvaam-Pronoun, same (gender)  
 Pary-verb Upaasate-aatmane padi, present, 3rd person, singular  
 asa-root.

**B. Output for Semantic Analyzer is as given below**

You entered string: mad bhaktah-my devotee You entered string: bhaktimaan-full of devotion You entered string: shraddadhaana-endued with shraddha You entered string: dharmaamrta- immortal dharma **Output for Derivative Analyzer is as given below:** You entered string: shaknosi.

The derivative is For present tense for parasmai padi Shaknositi Shaknositah Shaknosianti Shaknosisi Shaknosithah Shaknositha Shaknosiaami Shaknosiaavah Shaknosiaamah Similarly, for past, present future of parasmai and atmane padi is done.

**C. Output for Sandhi Analyzer is as given below**

Rule Based POS Tagger for Sanskrit

You entered string1: parama You entered string2: athma This is Deerga Sandhi: paramaathma

You entered string1: prakuthya You entered string2: eva This is Viruthi Sandhi: prakuthyaiva

**Test Cases:**

Table 1.6.1: Test Cases for POS Tagging

TEST CASES	INPUT	OUTPUT	RESULT
POS Tagging	mayi manah ye maam	mayi pronoun - same mayi manah noun manah, pratama vibhakti, singular, masculine mana ye pronoun-masculine ye pratma(saptami vibhakti, dual singular, masculine feminine-dual-pratama maam pronoun-same maam ditiya vibhakti, singular, masculine neuter ditiya vibhakti, singular, feminine	PASS
Semantic Analysis	Pata	Read	PASS
Derivative Analysis	Pata	All DERIVATIVES OF Pata in Present Tense: patati, patatah, patanti, patasi, patatah, patatha, pataami, pat aavah, pataamah	PASS
Sandhi Generator	Ganesha astakam	Ganeshaastakam	PASS

Accuracy Calculation: POS Tagging:

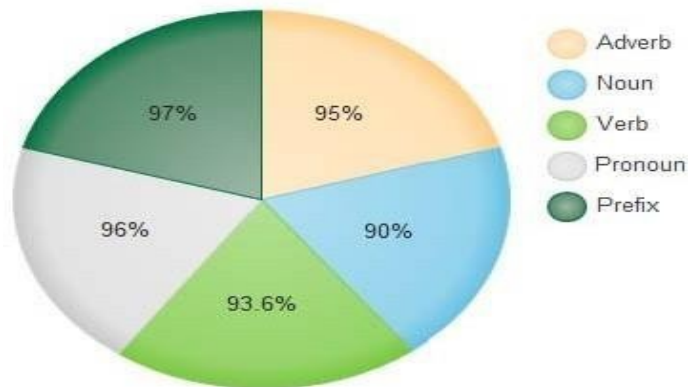


Figure 1.6.7.1: Pie Chart for POS Tagging



**Semantic Analysis**

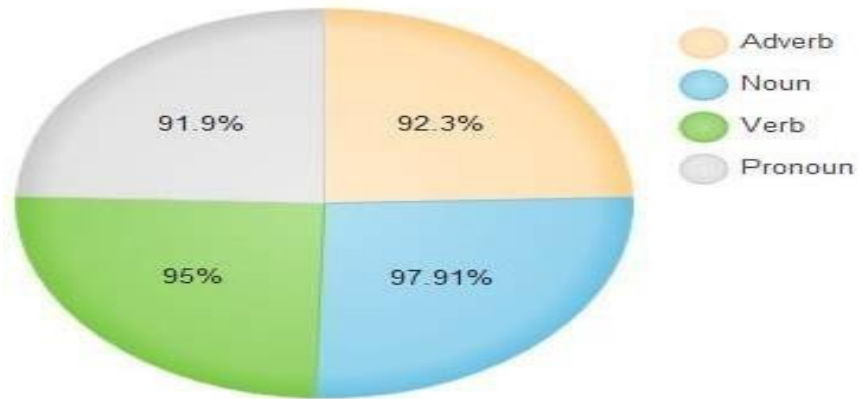


Figure 1.6.7.2: Pie Chart for Semantic Analysis

**Derivative Analysis**

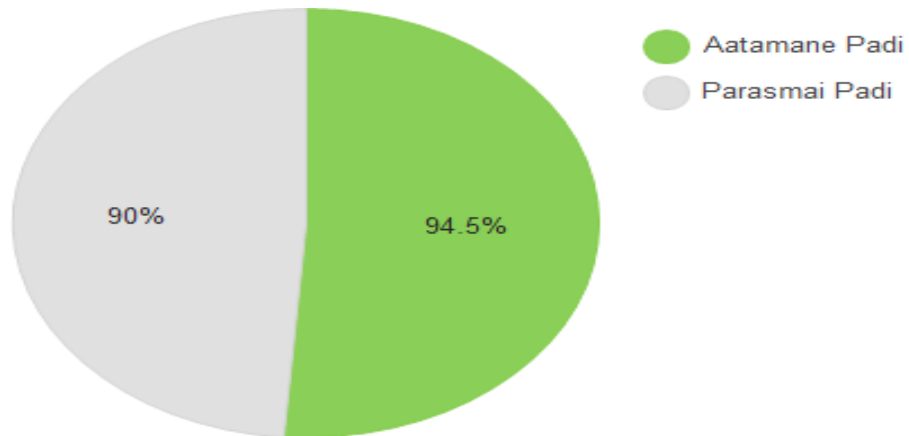


Figure 1.6.7.3: Pie Chart for Derivative Analysis

**Sandhi Generator**

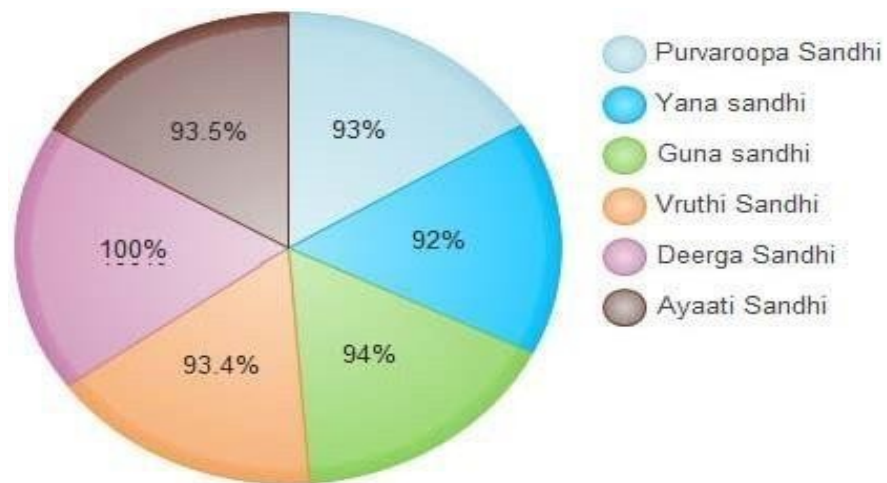


Figure 1.6.7.4: Pie Chart for Sandhi Generator

## ACKNOWLEDGMENT

We thereby thank the Vice Chancellor Prof. R.Sethuraman Dean, School of Computing, Dr.P.Swaminathan and Prof. Naren.J for extending their support and guidance in doing the Paper.

## REFERENCES

- [1] A.J.P.M.P. Jayaweera, N.G.J. Dias, "Hidden Markov Model Based Part Of Speech Tagger For Sinhala Language", *International Journal on Natural Language Computing (IJNLC)* Vol. 3, No.3, June 2014.
- [2] R Muni Prashanthi, M. Sirish Kumar, R.J. Rama Sree," POS Tagger For Sanskrit" *International journal of Engineering Sciences Research*, Vol- 04,(2013), ISSN:2230-8504; e-ISSN-2230-8512.
- [3] Pallavi Bagul, Archana Mishra, Prachi Mahajan, Medinee Kulkarni, Gauri Dhopavkar, "Rule Based POS Tagger for Marathi Text", (*IJCSIT*) *International Journal of Computer Science and Information Technologies*, Vol. 5 (2) , 2014, 1322-1326.
- [4] Kanak Mohnat, Neha Bhansa, Shashi Pal Singh, Ajai Kumar, "Hybrid approach for Part of Speech Tagger for Hindi", *International Journal of Computer Technology and Electronics Engineering (IJCTEE)* Volume 4, Issue 1.
- [5] <http://chandanasamskritam.blogspot.in>
- [6] <http://sanskrit.jnu.ac.in/post/post.jsp>
- [7] <http://nlp.stanford.edu/software/>.
- [8] J.P. Gupta, Devendra K. Tayal , Arti Gupta , "A TENGGRAM method based POS tagging of multi- category words in Hindi language ", *Expert Systems with Applications*, Volume 38, (2011),Pages 15084–15093.
- [9] Kh Raju Singha, Bipul Syam Purkayastha , Kh Dhiren Singha, "Part of Speech Tagging in Manipuri with Hidden Markov Model", *IJCSI International Journal of Computer Science Issues*, Vol. 9, Issue 6, No 2, November 2012, ISSN (Online): 1694-0814.