IJCRT Publish Paper record is available at DOI: http://doi.one/10.1727/IJCRT.17184

A Review on Implementation of Sandhi Viccheda for Sanskrit Words

Bhagyashree D. Patil[1]
M.E. Student
Computer Department, SSBT's COET, Jalgaon

Manoj E. Patil
Associate Professor
Computer Department, SSBT's COET, Jalgaon

*Abstract***Natural Language Processing is a field that covers computer understanding and manipulation of human language, and its ripe with possibilities for news-gathering. Sandhi means to join or combine two words to form meaningful word. Sandhi viccheda means sandhi splitting which breaks a word into its original word. Many projects have been implemented on concept sandhi viccheda in different languages such as Hindi, Urdu, and Kannada language but in Sanskrit there are many grammar rules which are not easy to implement. As we know Hindu religious book is BHAGWATGEETA containing most difficult words which a user can't understand easily due to longer words which are combination of different words. To overcome this problem there are some rules in Sanskrit such as rules regarding with vowels, consonants and visarga which should be implemented so that any word in Sanskrit can be separated. Till today only rules with vowels of sandhi are implemented. The accuracy will be increased by implementing the rules with consonant and visarga also. Accuracy is based on the correct linguistic rules and data provided to the system.**

Keywords— **Natural Language Processing, Sandhi Viccheda Rules.**

I. INTRODUCTION

Natural Language Processing is a field where one language can be converted into another using various approaches. Natural language processing is used for communication between computers and human languages in the field of artificial intelligence, and linguistics. Being concerned with human-computer interaction, NLP works to enable computers to make sense of human language to make interactions with machinery and electronics as user friendly as possible. Many more systems are developed for language translation like in Marathi, Telgu, Kannada, Punjabi, Hindi, etc. Sanskrit language has the most powerful grammar which gives the understanding of each word easily. The creator of Sanskrit language was Panini who formulated 3,949 rules. In NLP, not only words are there to understand but how they are linked together to derive a certain meaning is also important.

Sandhi means to join or combine two words to form meaningful word. Sandhi viccheda means sandhi splitting which breaks a word into its original word. Sandhi is a cover term for a wide variety of sound changes that occur at morpheme or word boundaries. Examples include fusion of sounds across word boundaries and the alteration of one sound depending on nearby sounds or the grammatical function of the adjacent words. Sandhi belongs to morph phonology. To develop a system for sandhi viccheda in Sanskrit is quiet difficult task because of its linguistic rules.

In order to achieve the main aim of sandhi viccheda process in Sanskrit language, rule based algorithm is used for the separation of words in its constituent words. For this, the words in Bhagwatgeeta are taken as a input to the system. In Bhagwatgeeta, there are many difficult words which a user can't understand easily without knowing their constituent words. So the main goal of this project is to provide a system to users such that user can get the meaning of difficult words in Bhagwatgeeta. The input is given to the system where it analyzes the word and precedes it towards the rule based algorithm where different rules according to the vowels, consonant and visarga are applied. Then it will be checked in the database to find the meaning of the word and the output is shown in the splitted words of original word with meaning.

Sandhi means to join or combine two words to form meaningful word. Sandhi viccheda means sandhi splitting which breaks a word into its original word. Many projects have been implemented on concept sandhi viccheda in different languages such as Hindi, Urdu, and Kannada language but in Sanskrit there are many grammar rules which are not easy to implement. As we know Hindu religious book is BHAGWATGEETA containing most difficult words which a user can't understand easily due to longer words which are combination of different words. To overcome this problem there are some rules in Sanskrit such as rules regarding with vowels, consonants and visarga which should be implemented so that any word in Sanskrit can be separated. Till today only rules with vowels of sandhi are implemented. The accuracy will be increased by implementing the rules with consonant and visarga also. Accuracy is based on the correct linguistic rules and data provided to the system.

The objectives of project are as follows:

- To split the complex words into its constituent words.
- To provide meaning of complex words after splitting for better understanding.
- To increase the accuracy of system by adding all rules regarding sandhi viccheda process.

II. LITERATURE SURVEY

- Amba Kulkarni, Sheetal Pokar, and Devanand Shukl [1] show that Verbal understanding (sabdabodha) of any utterance requires the knowledge of how words in that utterance are related to each other. Such knowledge is usually available in the form of cognition of grammatical relations. Generative grammars describe how a language codes these relations. Thus the knowledge of what information various grammatical relations convey is available from the generation point of view and not the analysis point of view. In order to develop a parser based on any grammar one should then know precisely the semantic content of the grammatical relations expressed in a language string, the clues for extracting these relations and finally whether these relations are expressed explicitly or implicitly. Based on the design principles that emerge from this knowledge, the parser for finding a directed Tree is derived, given a graph with nodes representing the words and edges representing the possible relations between them. Further, the Mimamsa constraint of akanksa (expectancy) to rule out non-solutions and sannidhi (proximity) to prioritize the solutions is used.

- Ms. Vaishali M. Barkade and prof. prakash R. Devale [2] proposed a system to develop a converter which converts English Statement to Sanskrit statement using rule based approach of machine translation. The proposed methodology uses a Rule based parser. The English sentence which is the input for first module i.e. lexical Parser it generates a Parse tree that is generated by using semantic relationships .This parse tree acts as an input to the Second module i.e. Semantic mapper where the English semantic word is mapped to the Sanskrit semantic word (Sanskrit word in English).

- Girish Nath Jha, Muktanand Agrawal [2] describes a Sanskrit morphological analyser which identifies and analyses inflected noun-forms and verb-forms in any given sandhi-free text.

- Ved Kumar Gupta, Prof. Namrata Tapaswi, Dr. Suresh Jain [3] shows knowledge representation of machine translation procedure of Sanskrit to English language by using the rule based approach. For this a parsing technique is used to generate Lexemes. Then it will be used in process of translation as input. Using some mapping rules and a dictionary based patterns desired output is produced for generating final results.

- Vimal Mishra, R.B.Mishra [4] integrates a Artificial Neural Network (ANN) model with traditional rule based approach of machine translation which translates an English sentence (source language sentence) into equivalent Sanskrit sentence (target language sentence). The feed forward ANN is used for the selection of Sanskrit word like noun, verb, object, adjective etc from English to Sanskrit User Data Vector (UDV). Due to morphological richness of Sanskrit language, the system makes limited use of syntax and uses only morphological markings to identify Subject, Object, Verb, Preposition, Adjective, Adverb and as well as Conjunctive sentences also. It uses limited parsing for part of speech (POS) tagging, identification of clause, its Subject, Object, Verb etc and Gender-Number-Person (GNP) of noun, adjective and object. This system represents the translation between the SVO and SOV classes of languages. The system gives translation result in GUI form and handles English sentences of different classes.

- The paper by Vimal Mishra and R. B. Mishra [5], gives a comparative view of EBMT and RBMT is presented on the basis of some specific features. The paper also describes the various research efforts on Example based machine translation and shows the various approaches and problems of EBMT. Salient features of Sanskrit grammar and the comparative view of Sanskrit and English are presented. The basic objective is to show with illustrative examples the divergence between Sanskrit and English languages which can be considered as representing the divergences between the order free and SVO (Subject-Verb-Object) classes of languages. Another aspect is to illustrate the different types of adaptation mechanism.

- Mrs. Namrata Tapaswi, Dr. Suresh Jain Mrs. Vaishali Chourey [6] proposed a paper which intends to introduce LFG (Lexical Functional Grammar) for parsing Sanskrit texts. The formalism of LFG has evolved from the extensive computational, linguistic, and psycholinguistic research, and hereby provides a simple set of grammatical rules for describing the common properties of all human languages and the particular properties of individual languages. The paper provides a set of instructions for using the formulation of LFG rules to parse Sanskrit. Also it can be useful for linguists who are unfamiliar with the formalism of the grammar for any language and they will find it possible to interpret and compose all the rules and lexical constructs that are standards in LFG. The paper presents successful parse of some simple sentences along with some unsuccessful parse of non-grammatical sentences. This verifies that the rules comply with the grammatical constructs of the language.

- Pawan Goyal, Vipul Arora and Laxmidhar Behera [7] proposed a concept of dependency parser for Sanskrit language that uses deterministic finite automata (DFA) for morphological analysis and 'utsarga apavaada' approach for relation

analysis is produced. A computational grammar based on the framework of Panini is being developed. A linguistic generalization for Verbal and Nominal database has been made and declensions are given the form of DFA. Verbal database for all the class of verbs have been completed for this part. Given a Sanskrit text, the parser identifies the root words and gives the dependency relations based on semantic constraints. The proposed Sanskrit parser is able to create semantic nets for many classes of Sanskrit paragraphs (Anuccheda). The parser is taking care of both external and internal sandhi in the Sanskrit words.

- Gerard Huet [8] presents the state of the art of a computational platform for the analysis of classical Sanskrit. The platform comprises modules for phonology, morphology, segmentation and shallow syntax analysis, organized around a structured lexical database. It relies on the Zen toolkit for finite state automata and transducers, which provides data structures and algorithms for the modular construction and execution of finite state machines, in a functional framework. Morphemes are assembled through internal sandhi, and the inflected forms are stored with morphological tags in dictionaries usable for lemmatizing. These dictionaries are then compiled into transducers, implementing the analysis of external sandhi, the phonological process which merges words together by euphony. This provides a tagging segmenter, which analyses a sentence presented as a stream of phonemes and produces a stream of tagged lexical entries, hyperlinked to the lexicon. The next layer is a syntax analyser, guided by semantic nets constraints expressing dependencies between the word forms. Finite verb forms demand semantic roles, according to valency patterns depending on the voice (active, passive) of the form and the governance (transitive, etc) of the root. Conversely, noun/adjective forms provide actors which may fill those roles, provided agreement constraints are satisfied. Tool words are mapped to transducers operating on tagged streams, allowing the modelling of linguistic phenomena such as coordination by abstract interpretation of actor streams. The parser ranks the various interpretations (matching actors with roles) with penalties, and returns to the user the minimum penalty analyses, for final validation of ambiguities. The whole platform is organized as a Web service, allowing the piecewise tagging of a Sanskrit text.

- Rupali Deshmukh Varunakshi Bhojane [9], the paper gives a survey of various sandhi splitting techniques for different Indian languages where sandhi splitting means the process by which one word is broken into its constituent words. Also sandhi is a process in which two or more words are unite to form a single word.

- Rupali Deshmukh Varunakshi Bhojane [10], Natural language processing (NLP) is a field of computer science, artificial intelligence, and linguistics, concerned with the interactions between computers and human (natural) languages. The conjunction of two immediate sounds, that means union, is called as sandhi formation. Sandhi-splitting describes the process by which one letter is broken to form two words. Sandhi-splitting is one subtask for complete analysis of input text in NL. Proposed system is an enhanced version, which recognizes sandhi word from input text in Sanskrit; splits sandhi word into its original form and return what type of sandhi it is. In future sandhi splitting can be done in Sanskrit for complex words using different approach.

- Vaishali Gupta, Nisheeth Joshi, Iti Mathur [11], Urdu is a combination of several languages like Arabic, Hindi, English, Turkish, Sanskrit etc. Here a stemmer is used to convert a word to its root form. The suffix and prefix are removed from the word to extract the actual word from it. The accuracy of the system is upto 85% but there is drawback which is over stemming and under stemming. Likewise in Urdu, we can implement the system for Sanskrit language also.

- Priyanka Gupta, Vishal Goyal [12], sandhi literally means `putting together' or combining (of sounds). It denotes all combinatory sound-changes affected (spontaneously) for ease of pronunciation. Sandhi-viccheda describes the process by which one letter (whether single or cojoined) is broken to form two words. Part of the broken letter remains as the last letter of the first word and part of the letter forms the first letter of the next letter. Sandhi-Viccheda is easy and interesting ways that can give entirely new dimension that add new way to traditional approach to Hindi Teaching. The Rule based algorithm proposed here give an accuracy of 60-80% depending upon the number of rules to be implemented. In future, Sandhi Viccheda can be implemented for different languages such as Sanskrit using rule based algorithm.

- Aasish Pappu and Ratna Sanyal [13], developing a Machine Translation system for ancient languages is much more fascinating and challenging task. A detailed study of Sanskrit language reveals that its well-structured and finely organized grammar has affinity for automated translation systems. In this paper necessary analysis of Sanskrit Grammar in the perspective of Machine Translation is provided and also provides one of the possible solutions for Samaas Vigraha (Compound Dissolution). The future scope decides that the complex words or the poems which conveys more than one meaning can be processed in MT system to increase the accuracy of system.

- Joshi Shripad S. [14], Sandhi splitter is an important module for Natural Language (NL) system for Marathi in which words can be combined together to form a larger string of words. The research in Natural language processing is being carried out in variety of areas like speech processing, text analysis, text processing, text mining etc. Among all there is a need of analysis of word in the given language. The formation of word may be the result of combination of two or more words. Separation of the words in meaningful sub-words is sandhi splitting (Sandhi-Viccheda). In this paper the rules

and the rule based algorithm for sandhi splitting of Marathi compound words are represented. In future, the same system can be implemented for Sanskrit language.

- M. Rajani Shree, Sowmya Lakshmi, Dr. Shambhavi B.R [15], Sandhi is also called Morphophonemics concerned with changes that occur when two words or separate morphemes come together to form a new word. Exact splitting point is essential for text processing tasks such as POS tagging and in turn parsing. A novel approach to internal Sandhi splitting technique on Kannada language is adopted. Each Kannada word is split into morphemes according to valid morph patterns. After the division of each word into lexical morphemes, each split word into root-begins, root-continuous and suffix has been manually tagged. The system with a list of 1000 tagged words using a CRF (Conditional Random Fields) tool and nearly 400 raw split words (untagged words) are given as input to the CRF tool. The system generates a list of tagged split words for the given input according to the trained data. The system output has been compared with the manually tagged data. The data has been verified using 5 fold test, which takes five different combinations of trained data (1000 words) and test data (400 words). The average Precision, Recall and F-Measure of Tagging accuracy of CRF model for Kannada corpus in 5 fold tests are nearly equal to 98.08, 92.91 and 95.43. This method can be successfully implemented in all other Dravidian languages for the Sandhi splitting. The future work can be done with Sanskrit language to split the words using the different approach.

- Sachin Kumar [16], presents the sandhi splitter and analyser for Sanskrit language. The analysis procedure of the system uses lexical lookup method as well as rule base method. Before sandhi analysis process, pre-processing, lexical search of sandhi strings in sandhi example base and subanta-analysis takes place respectively. The pre-processing will mark the punctuation in the input. After that, the program checks the sandhi example base. This example base contains words of sandhi-exceptions (vārttika list) and commonly-occurring sandhi strings (example list) with their split forms. These words are checked first to get their split forms without parsing each word for processing. After lexical search, subanta analyser gets the case terminations (vibhakti) separated from the base word (prātipadika). Subanta analyser also has a function to look into lexicon for verb and avyaya words to exclude them from subanta and sandhi processing. The subanta analysis will be helpful in the validation of the split words generated through reverse sandhi analysis as the Sanskrit words in lexicon are stored in prātipadika form. The reason to accumulate the words in prātipadika form is that sandhi-derived words in input Sanskrit text may have any of the case terminations. After subanta-normalization of input text, the system will look for fixed word list of place name, nouns and MWSDD. The words found in these resources will be let off form processing.

- Manji Bhadra, Subhash Chandra, R. Chandrasekhar and Sudhir K Mishra [17] proposed a system called as SAS (Sanskrit Analysis System)a complete analysis system for Sanskrit. Some modules of this system have already been developed. The system accepts full text inputs in Devanagari Unicode (UTF-8). The sandhi module does the segmenting for complex tokens and then hands over the text for detailed processing. The SAS has two major components - the shallow parser and the karaka analyzer. The shallow parser has separate modules, some of them are partially implemented, and some of them are in the process of being implemented. The modules have been developed as java servlet in Unicode using RDMBS techniques. The applications of the SAS will be many ranging from being a Sanskrit reading assistant to machine translation system from Sanskrit to other languages.

From the above literature survey, various method and techniques are used for sandhi viccheda of words. But for complex words there are some exceptions. For that the system proposed will work and will give the accurate sandhi viccheda module to understand the words easily. Let us see the existing system.

III. EXISTING SYSTEM

Some research work has been done related to Indian languages. There are some existing systems available which can split sandhi words and generates of which type it is. Some existing systems are listed below in table 3.1.

Basis	Sanskrit Analysis System [17]	The Sandhi Engine [9]	Sanskrit Splitter and Analyzer [16]	Vowel Sandhi Viccheda System [10]
Features	System takes input from keyboard or iTRANSDevanagari Unicode converter. System gives output with sandhi type. This system is for sandhi splitter as well as sandhi generator.	The input window may accept Unicode input under UTF-8 encoding, either in Devanagari, or in Indological Romanization script. This system is for sandhi generator only.	The input is given in Devanagari language means in sanskrit Language. Generates possible all solutions for given input.	The input is given as the Sanskrit text or sentence. Using maker they have generated the output of split.
Approach	Rule based and Example based	Rule based	Rule Based Method	Rule based approach

Advantages	Using Paninian Rules for generating reverse computation of sandhi rules.	The engine takes input in roman transliterate on and returns output in roman as well as Unicode script.	It takes input in Sanskrit language and provides an appropriate solution for that.	Using reverse sandhi rules splitting is done. Sentences are given as input.
Limitations	It works only for vowel sandhi splitting. Sometimes it gives multiple results at a time.	Generation of words based on ad-hoc processing and not using Paninian rules. Not giving specific type of sandhi in output.	Recognition process for rule base is difficult also there some problems regarding the rules for vowels, consonants i.e. exceptions in rules which are usually unimplemented.	Only vowel sandhi system is implemented.

Table 3.1 Existing systems

IV. CONCLUSION

Sandhi Viccheda play an important role in Sanskrit because using that a user can understand the actual word. Many researchers have developed a system for sandhi viccheda module for different languages. Also they have used different approaches to implement sandhi viccheda module. Sandhi viccheda system requires more correct linguistic rules for implementation.
In this paper, a review has been taken on sandhi viccheda in Sanskrit. Also some existing systems are discussed above which has their own advantages, features and limitations. Up till the vowel sandhi splitter module has been proposed, but the system accuracy will be increased by adding the rules regarding consonants and visarga also.

REFERENCES

[1] S. P. Amba Kulkarni and D. Shukl, “Designing a constraint based parser for Sanskrit," SpringerLink, Sanskrit Computational Linguistics pp 70-90, 2010.
[2] P. P. R. D. Ms.Vaishali M. Barkade, “English to Sanskrit machine translator," International Journal on Computer Science and Engineering, vol. 02, 2010.
[3] P. N. T. Ved Kumar Gupta and D. S. Jain, “Knowledge representation of grammatical constructs of sanskrit language using rule based sanskrit language to English language machine translation," International Conference on Advances in Technology and Engineering (ICATE), IEEE, 2013.
[4] R. M. Vimal Mishra, “ANN and rule based model for English to Sanskrit machine translation," ResearchGate, August 2017.
[5] V. Mishra and R. B. Mishra, “Study of example based English to Sanskrit machine translation," June 2008.
[6] D. S. J. Mrs. Namrata Tapaswi and M. V. Chourey, “Parsing Sanskrit sentences using lexical functional grammar," International Conference on Systems and Informatics, IEEE, 2012.
[7] V. A. Pawan Goyal and L. Behera, “Analysis of Sanskrit text: parsing and semantic nets," Springerlink, Sanskrit Computational Linguistics pp 200-218, vol. 5402, 2009.
[8] G. P. Huet, “Shallow syntax analysis in Sanskrit guided by semantic nets constraints," ResearchGate, 2017.
[9] V. B. Rupali Deshmukh, “Sandhi splitting techniques for different Indian languages," International Journal of Engineering Technology, Management and Applied Sciences, vol. 2, December 2014.
[10] V. B. Rupali Deshmukh,“Building vowel sandhi viccheda system for sanskrit," International Journal of Innovations and Advancement in Computer Science, vol. 4, December 2015.
[11] I. M. Vaishali Gupta, Nisheeth Joshi, “Rule based stemmer in Urdu," 2013 4th International Conference on Computer and Communication Technology, IEEE, 2013.
[12] V. G. Priyanka Gupta, “Implementation of rule based algorithm for sandhi-viccheda of compound Hindi words," 2009.
[13] A. Pappu and R. Sanyal, “Vaakkriti: Sanskrit tokenizer," January 2008.
[14] J. S. S., “Sandhi splitting of Marathi compound words," International Journal on Advanced Computer Theory and Engineering (IJACTE), vol. 2, 2013.
[15] S. L. M. Rajani Shree, “A novel approach to sandhi splitting at character level for kannada language," 2016 International Conference on Computational Systems and Information Systems for Sustainable Solutions, IEEE, 2016.
[16] Sachin Kumar, “Sandhi splitter and analyzer for Sanskrit (with special reference to ac sandhi)," 2007.
[17] R. C. Manji Bhadra, Subhash Chandra and S. K. Mishra, “Sanskrit Analysis System (SAS)," ResearchGate, January 2009.