

**LANGUAGE IN INDIA**  
**<http://www.languageinindia.com>**  
**Volume 5 : 1 January 2005**

**SANSKRIT TO ENGLISH TRANSLATOR**  
**Aparna Subramanian**

---

**Contents**

<a href="#"><u>Certificate of Approval</u></a>
<a href="#"><u>Acknowledgement</u></a>
<a href="#"><u>CHAPTER 1. INTRODUCTION</u></a>
<a href="#"><u>CHAPTER 2. ANALYSIS</u></a>
<a href="#"><u>CHAPTER 3. OVERALL DESIGN</u></a>
<a href="#"><u>CHAPTER 4. CASE STUDY</u></a>
<a href="#"><u>CHAPTER 5. CONCLUSIONS</u></a>
<a href="#"><u>REFERENCES</u></a>

---

**CERTIFICATE OF APPROVAL**

This is to certify that the dissertation entitled "Sanskrit to English Translator" submitted by Miss Aparna Subramanian for the partial fulfillment of the Degree of the Master of Science in Computer Science from the School of Computer Science, Devi Ahilya Vishwavidyalaya, Indore is a satisfactory account of her project work and recommended for the award for the degree.

Dr. A.P. Khurana  
Head, S.C.S  
DAVV, Indore

---

## Acknowledgement

I owe the success of the project to several scholars and friends, some directly guided me in building the project, and others motivated me indirectly. Dr. M.Chandwani, Director, I.E.T, at the Paradigm 2k2 workshop conducted by SCS students, sowed the seed of interest in Natural Language Processing in me a year ago. Had it not been for the seminar given by Dr. Chandwani, I would have never been able to unravel the immense interest in this field that was dormant in me. Then I went up to Mr. Suresh Jain, Reader, I.E.T., who had been teaching our Theory of Computation classes, with the idea of translation between Sanskrit and English. He showed a keen interest in this proposal, thus inspiring me to work on machine translation. Then on, I was constantly engaged in gathering materials and ideas on this subject. But concrete work in this direction did not start until Dr. M Ingle encouraged me to present a technical paper on morphological parsing at I-Maze international conference. Since then, she has been guiding me, and helping me with materials, and valuable information for the project.

I am indebted to Dr. Sharma, retd. Principal of Kendriya Vidyalaya, Chennai, Who helped me build a clearer view of Sanskrit grammar and provided me with relevant books concerning this idea. I am also grateful to the many Sanskrit scholars in Indore and Chennai, from whom I sought help, and also to Professors in IIT Kanpur who encouraged me in building this translator.

I would like to thank all the authors of the different books that I used and papers that I consulted. I am also thankful to many others who indirectly played a part in the successful completion of the project.

**Aparna Subramanian**

---

## CHAPTER 1 INTRODUCTION

### 1.1 Preamble

The present project concerns with the Machine Translation domain of Natural Language Processing. This area of Artificial Intelligence is very useful in providing people with a machine, which understands diverse languages spoken by common people. It presents the user of a computer system with an interface, with which he feels more comfortable. Since, there are many different languages spoken in this world, we are constantly in need for translators to enable people speaking different

languages to share ideas and communicate with one another. Human translators are rare to find and are inaccessible to the common man. With the concept of Machine

Translation we may work towards achieving the goal using easily available computer systems. Besides, one of the first linguistic applications of computers to be funded was machine translation.

The Sanskrit to English translator designed will be very useful to people in India in particular and the world in general. The kinds of computer languages best suited for this purpose has been found to be those based on logic, like PROLOG. It will be a great attempt towards bringing people together if this project is successfully completed. In the subsequent subsections, we give a detailed introduction to the project describing the need for the project and how the project is proposed to look like.

### **1.2.1 Developments To Present Date**

There has been a lot of interest shown in the field of Machine Translation from as early as the 1950s. Early attempts by the military and intelligence communities in the US during this period amounted to no more than word-by-word translations. Later though, there has been a lot of progress in Machine Translation, and there has been renewed interest since the early 1980s. Programs like METEO were developed. These have been used to translate weather forecasts from English to French in Canada since 1977. Other large noteworthy efforts were: Eurotra for European languages, Mu for Japanese and English, and KMBT for translation between English and Japanese developed at Carnegie Mellon University, etc. Many more attempts have been made on translation of Indian languages.

India too has active groups in Machine Translation. The earliest published work was by Chakraborty in the year 1966. More recently, work has been undertaken in Tamil University in Thanjavur, National Center for Software Technology (NCST) in Mumbai, Center for Development of Advanced Computing (C-DAC) in Pune and IIT Kanpur. A research group at Thanjavur attempted Russian to Tamil translation based on the direct approach in 1985 and the first system translated simple sentences. A group at NCST did some work on English to Hindi translation but did not develop a working system. Translation from Hindi to English is going on at the IIT Kanpur, which is being funded by the Indian Government. However, except for some attempts concerning processing of Sanskrit texts performed by C-DAC, actual attempt towards translation from Sanskrit to English has not been made till date. An idea towards designing such a system was a paper I presented at the I-Maze international conference along with Dr. (Mrs.) Maya Ingle. It discussed the Sandhi Vichcheda part

of the translation process, which is implemented in totality in the present project. The Project also enhances this idea to perform the actual translation.

### **1. 2. 2 Existing Need**

People both in India and abroad are surely and steadily realizing the importance of ancient scripts in the diverse fields of science, commerce and arts. Also, as spiritual awareness sweeps the world, great efforts are being made to present the vedic scriptures in different languages to the common people. In IIT Kanpur, a group of professors and students is involved in a noble cause of putting up Shrimad Bhagavad Gita and its translation on to the internet. The vedic texts mentioned above are mainly in the Sanskrit language.

Since Sanskrit is an ancient language and is no more in vogue as an easily understandable language, many communities are constantly working towards the translation of Sanskrit texts to popular languages.

Presently, scholars of Sanskrit are doing this translation work manually and they have expressed a need for some software to do this work for them.

The software that is modeled here is an attempt to ease the work of scholars and help accelerate the translation efforts. It shall also make it possible for the common people, who are not familiar with Sanskrit, to translate texts and understand them.

The most popular language in the world today is English. So, a translator which can translate Sanskrit sentences into English will prove very useful. As already mentioned in the previous section, a Sanskrit to English translator has not yet been developed. The proposed software is an effort to develop this for use in a restricted domain.

### **1. 2. 3 Objectives**

As the preceding subsection tells us, there is a need for a translator to translate sentences from Sanskrit to English. The project built here presents a model for this purpose. In this subsection, we describe what is within the scope of the project and what is not. Broadly, we may identify the two main objectives of the present project as performing sandhi vichcheda of Sanskrit words and then translating them to English.

Since, conjunction of individual words of Sanskrit into longer ones is a common occurrence in Sanskrit texts, sandhi vichcheda is the most important part of the translator. The objectives for this stage are the following:

1. 'Sandhi Vichcheda' should include Vichcheda of the vowel conjunctions. Though the software may be further extended to include Vichcheda of consonants also, it has not been considered at the moment.
2. While performing vichcheda, the module must also analyze its constituent words. The analysis should lead to a parse of the individual words specifying the part of speech, gender, number, tense etc., whichever of these is applicable.

There are some important aspects of translation that have been considered in the present system. They are enumerated below:

1. The syntactic arrangement of the sentence. Since, Sanskrit is a free structured language as far as formation of sentences are concerned whereas English follows the famous SVO (Subject Verb Object) order of sentence formation, it becomes necessary to give translated sentence in the correct form.
2. When translated, Sanskrit words may correspond to more than one phrase of English. The most dominating of these cases is that of nouns, where the translated phrase needs to be selected among different phrases formed due to the possibility of using different prepositions for the same noun case. The objective of the present system is to deal with these situations by using the most popularly used prepositions in each case.

Apart from the two main objectives described above, the other features of the project are:

1. 'Simple Sanskrit texts' will involve a few number of nouns belonging to the common categories of Akarantha, Ikarantha, Ukarantha etc. of the feminine, Masculine and Neuter genders.
2. A few common pronouns of the above categories shall also be included.
3. Commonly used Parasmaipadi verbs in simple present tense has been included.
4. Input is in the form of Roman literals, thus eliminating the need for transliteration while programming and enabling us to concentrate on the logics of translation.
5. The sentences are given one at a time.
6. The output checks the validity of the sentence as well as gives the appropriate translation only if it is valid.

As we are now clear about the objectives, let us explore the probable solutions to the problem.

## **1. 2. 4 Possible Solutions**

The objectives described above may be implemented in numerously many ways. One of these has been employed for the present project. In this subsection we identify some of the important approaches and briefly describe why the present approach has been chosen.

The separate lexicons for Sanskrit and English may be maintained in a database with morphological details stored in the form of logics in the programming language used. A bilingual dictionary will also have to be maintained in this case.

Another approach may be to store a single bilingual dictionary in the form of a database, which also describes the morphological details and write programs in a high level language to access this huge database and translate the sentence accordingly.

In both the cases, grammar also needs to be stored separately in the form of implemented logic. In both the above approaches a combination of databases along with logic in some structured or object oriented programming language is suitable. The present work takes a slightly different both for representing the rules as well as the knowledge base. An appropriate programming language in the form of Visual prolog has been taken to implement the approach. The system stores the detailed morphological knowledge in the form of finite state transducers, which are nothing but implemented Moore machines. The lexicon is a set of facts. The grammar and the translation logic are stored as rules. The details of the design are given in chapter 2.

The methodology used in this project is different to the other two described in the beginning of the section in the sense that the morphological details as well as the lexicon are stored together. In the other two, the lexicon had to be stored separately and the logic for morphological analysis had to be written separately. Also, the present system possesses easy parsing capabilities. Therefore, it is superior to the other plans introduced.

## **1. 2. The Aim of the Project**

The conception of the idea of building the present project had purely research motives. Though translation of sentences from Sanskrit to English is the main aim of the present project, performing sandhi vichcheda and performing the morphological parsing simultaneously receives a greater part of the attention in building the system. The aim was to implement the idea presented in my paper titled 'Morphological parsing of sandhi based words in Sanskrit texts' and extend it to perform the translation of the sentences too.

The aim of this project is defined below:

To translate simple Sanskrit texts involving the need for Sandhi Vichcheda into corresponding appropriate sentences in English.

The aim of translation defined above needs to be fulfilled keeping in mind the objectives defined in section 1.1.3. In the above definition, the phrase 'simple Sanskrit sentences' indicates that the lexicon is small and has a limited domain of commonly used words, where, new words can be incorporated as and when required. Also, since resolving semantic ambiguities has been kept out of the scope of this project, sentences need to be semantically sound. Semantic errors would lead to incorrect results, though syntax errors would be notified.

---

## CHAPTER 2 ANALYSIS

### 2. 1. Morphological Parsing and Sandhi Vichcheda

In the machine translation of Sanskrit sentences, the primary concern of translation is the morphological parsing of the words of Sanskrit. Since, the words are a combination of more than one basic word, they have to be subdivided, and the subdivided words need to be parsed. The morphological parsing gives us total information of the class to which the separated words belong. The parsing of Sanskrit words is necessary because the parsed input determines what phrases will replace the Sanskrit text when it is translated into the English language. Translation involves generation where proper ordering of the sentences must be taken care of. There are some ambiguity issues involved in translation. One of the main issues as far as the translation from Sanskrit to English is concerned is the choice of prepositions.

#### Morphological Parsing

The morphological parsing of words belonging to natural languages involves, providing a structure of the given input indicating the different morphemes that constitute the input and how they are related to each other. Morphemes are smaller meaning bearing units of words. These morphemes can be broadly classified into stems and affixes, which may be prefixes, suffixes, infixes or circumfixes. While prefixes are those morphemes, which may appear before a stem, and postfixes are those that are applied to the end of the stem, circumfixes are those morphemes that may be applied on both sides of the stem. The morphemes categorized under infixes are those that appear inside a stem.

Any morphological parser must consist of parts acting as a lexicon giving an exhaustive list of all the morphemes present in the language, the morphotactic rules specifying the position of one morpheme with respect to another and a set of orthographic rules which specifies how two given morphemes combine. The lexicon combined with the morphotactics may be modeled using a finite automaton. The finite automaton may be a transducer of some kind. Orthographic rules may be used at a stage when it is needed that the morphemes be combined together to give the surface word.

### **Example 1:**

In a word like 'books'- 'book' and '-s' are the two morphemes. When the word 'books' is parsed, the parsed output tells us that the word was formed because a noun called 'book' had been used in its plural sense, i.e. the output could be something like- (book, Noun, Plural). This information can later be used to translate the word 'books' to other languages.

### **Sandhi Vichcheda in Sanskrit**

The inputs to the parser are words, which may further be a merger of more than one Sanskrit word. This kind of systematic blend of words is called sandhi. The Sanskrit word sandhi means 'to join together'. Sandhi is a coalescence of two letters coming in immediate contact with each other. When two words are adjacent to each other, the last letter of one word and the first letter of the next word come in immediate contact and may lead to a unification of the two words to form a larger word. This unification takes place according to the rules of sandhi. Practically, a word formed by sandhi rules may contain any number of smaller words.

Broadly, the sandhi types may be classified as Swarasandhi and Halsandhi. Swarasandhi deals with vowel combinations, while Halsandhi deals with combinations of consonants.

### **Example 2:**

- a) a+I=E as in sva+IraH=svEraH ( One acting willfully)
- b) a+A=A as in rAmeNAnIwaH=rAmeNa+AnIwaH (Brought by Rama)

### **Parsing of Sandhi Based Words**

The morphological parsing of each basic word can be done along with the vichcheda. Basic words are generated one at a time from long coalesces of many words. Each

word now may belong to any of the parts of speech. We now discuss the peculiarities of each type of word that needs to be parsed. The nouns in Sanskrit may belong to one of the genders- male, female, neuter and may further be Aakaarantha, Ikaaraantha, Ukaaraantha, etc. according to the ending letter of each basic words. We will also need to recognize the categories of each noun according to the case and number.

There are seven cases in Sanskrit. The numbers are singular, dual and plural. Similar categories exist for all pronouns. Verbs and their various forms also need to be differentiated. Verbs are broadly characterized into parasmaipadi and aathmanepadi words. There are twelve different tenses each one of them falling into one of the three tenses namely, present tense, past tense and future tense. Verb forms in each tense are either first person second person or third person, and may change according to the number, which may again be single, dual or plural. All other words fall into the category called the indeclinable (avyayas). All the above categories are to be recognized and the parse of the word needs to give these information.

## **2. 2. Translation**

The correct translation of the sentence largely depends on the parse that is obtained for it. So, depending of the language in which it has to be translated, the details of the parse of each word should be designed appropriately.

## **Generation**

Once the parse gives the details of each word, the corresponding phrase in English needs to be generated. An important aspect of translation is the grammatical structure of the sentences. Different languages follow varied syntactical structures. While Sanskrit follows a free ordering of sentences, English sentences follow SVO (Subject Verb-phrase Object) format for active and OVS (Object Verb-phrase Subject) format for passive sentences. For example, 'rAmaH puswakaM paTawi' which is translated as 'Rama read the book' may also be written as 'puswakaM paTawi rAmaH' or 'paTawi rAmaH puswakaM' or 'rAmaH paTawi puswakaM' or 'puswakaM rAmaH paTawi' or 'paTawi puswakaM rAmaH'.

## **Ambiguities**

Each individual word in Sanskrit corresponds to a word or a phrase in English. Also, There are many syntactically correct equivalent phrases of English corresponding to each word of Sanskrit. Resolving this ambiguity needs extra knowledge categorizing words of the sentence according to their peculiarities. When we deal with nouns of Sanskrit, corresponding phrases of English are nouns of English along with

appropriate prepositions. The prepositions accompanying each noun depend on the case.

For each case in Sanskrit, there is a set of possible prepositions. Resolving this ambiguity needs a thorough study of individual words and word categories in general. For example, in the sentence 'saH xaNdena wAdayati', xaNdena when taken individually, can be translated as either 'by stick' or 'from stick' due to the nature of the noun being in the accusative case. But we know from experience that it is more reasonable to consider the probability of occurrence of something done 'by' the stick than something being obtained 'from' the stick.

So, three approaches to solve this problem would be - to consider the preposition with higher probability of occurrence for all nouns of Sanskrit, or to store knowledge of the appropriate preposition for each noun, or to determine the most suitable preposition from a combination of the word knowledge and the semantics of the sentence.

---

## CHAPTER 3 OVERALL DESIGN

### 3. 1. Introduction

The careful analysis of the project and its possible solutions leads to the following design of the system. The sections to follow give the total design as well as the details of the design of individual modules. The details are illustrated using diagrams and examples.

The system consists of a morphological parser and a translation generator. The parts of each of these two are described in detail in this chapter.

### 3.2 System Design

The overall system is diagrammatically shown below:

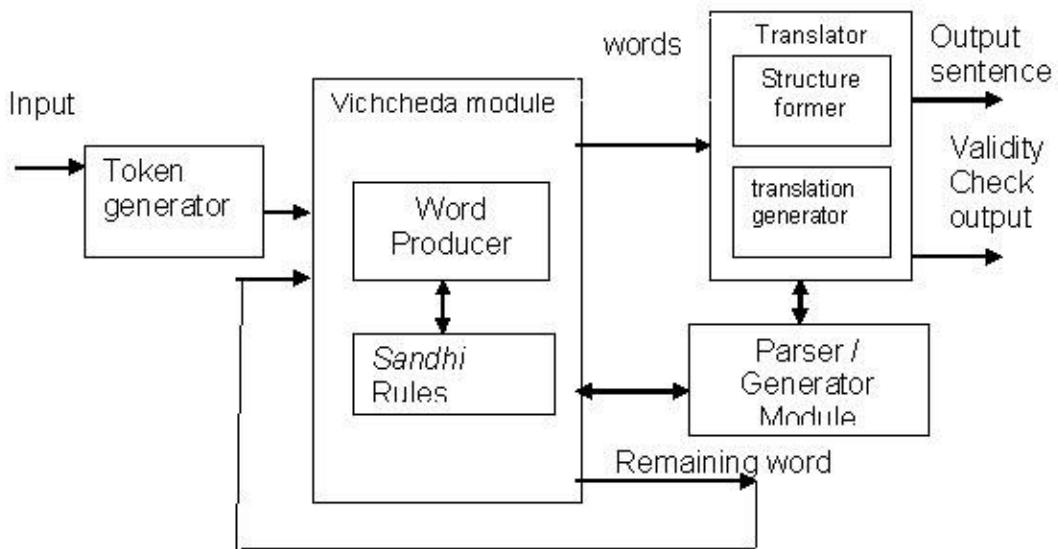


Figure 1  
**The complete design of the translator**

## Token Generator

This module splits the given sentence into chunks of strings delimited by spaces. These strings may be simple words or compound words coalesced by the rule of Sandhi.

## Vichcheda Module

The vichcheda module gets help from the set of transducers to identify words and forms words through the word generator. The word generator in turn takes the help of the sandhi rules module wherever necessary. The remaining string after the basic word is generated is sent back to the vichcheda module.

## Translator

This module performs the actual translation. The input to this module is the parse. It also interacts with the parser/generator module to get the parse of each word. It then generates appropriate equivalents in English for the morphological details of each word and ultimately presents the sentence in the correct order.

## Parser Generator Module

This module contains a set of transducers built for individual Sanskrit words and transforms strings to partial words, which are used by the vichcheda module. It also gives the parse of the words, which are used by the sentence former to give the output in a structurally correct sentence.

### **3. 3 Detailed Design**

As discussed earlier, the translation from Sanskrit to English takes place in two phases namely, the morphological parser and the translator module. The input to these modules should be continuous string of characters. The 'sentence splitter', as seen in the previous section, achieves this. The morphological parser consists of a set of transducers that transform the given input to a set of acceptable. They also give the parse for these words. The second part of the parser is a Vichcheda module, which splits the input to the basic words of Sanskrit. 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.

While parsing words formed using orthographic rules of sandhi, there is a need to generate all the set of sub-words contained in each input word along with the parsed output of each such sub-word. The parser/generator helps us achieve this. The Parser/Generator module uses a part of the input and generates the outcome of a split. Another string, which is also generated by the Parser/ Generator is used by the Vichcheda module. The Vichcheda module then converts this to an appropriate string, which confirms to the sandhi rules. It also appends a string to the beginning of the remaining string obtained after splitting. The remaining string is given back to the system as input. This process is repeated until the input word has been totally parsed.

The different modules are separately described below.

#### **Sentence Splitter**

The work of this module is very simple. It groups the string of continuous characters together. Its means of finding this is the space character or a set of space characters. The strings thus got are not necessarily isolated words of Sanskrit. They may be a set of concatenated words that need to be broken down.

#### **Parser/Generator Module**

This module is a collection of transducers as described before. The transducers are Moore Machines with multiple outputs. The transducers are defined as:

1. A finite set of states  $s_0, s_1, s_2 \dots$  where  $s_0$  is the initial state.
  2. An alphabet  $A = (\{\text{devanagri letters}\} \cup \text{null})$ , for input strings and two sets of output strings.
  3. An output Alphabet  $O \subset C$  ( $C = \{\text{part of speech, case, number, gender, tenses, ...}\} \cup \text{null}$ ).
  4. An output table which gives the three sets of outputs.
  5. A pictorial representation that represents the transitions between states.
- A general representation is as shown in figure 2 below.

All transducers are combined to form a morphological parsing system module called the Parser/Generator. The design of this set of transducers is given in figure 3 where each transducer is of the type of figure 2. There is one transducer for each word, which takes care of its declination if applicable. The declination part is handled by adding the appropriate suffixes (obtained from the knowledge of the Sanskrit declination tables) to the stem. This job is taken care by the transducer. The input to the module is directed to the transducers. The transducers parse the input letter by letter. When enough input letters have been read that indicates some possible grammar characteristic, it is indicated in the 'parse string' output. At each state, an output corresponding to the string, i.e. 'partial root' is stored in the output table. This is sent to the Vichcheda module. Figure 3 shows that the knowledge is stored as a set of transducers. There is a transducer for each type of a word.

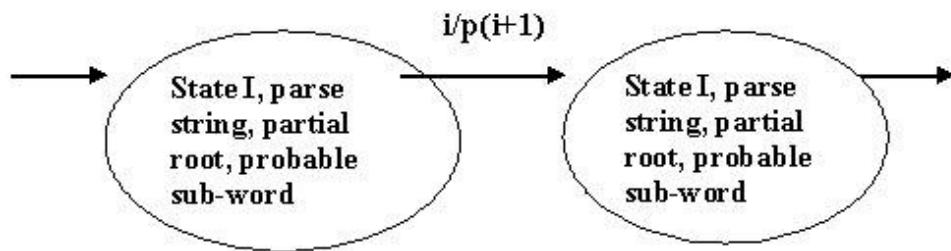
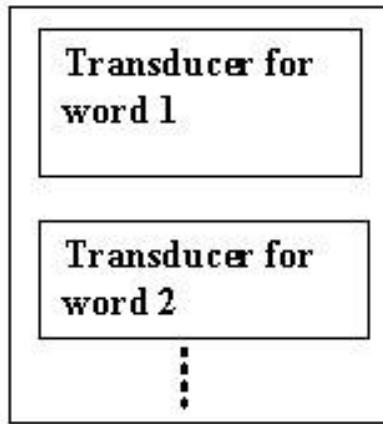


Figure 2  
Transducer implemented as a Moore machine



**Figure 3**  
**Set of transducers each for a set of similar words**

### **Example**

The Sanskrit word rAmasyAcAryah (Acharya of Rama) is a coalescence of the words 'rAmasya' (of gods) and 'Acharyah' (Acharya or teacher). The transducer will parse 'rAmasya' first and then parse 'Acharyah' in the next turn. The transducer for 'rAma' is given as an example in figure 4. When the transducer gets an input rAm, it immediately recognizes the word to be a proper noun denoting a name of a person. Next, the substring 'as' indicates that rAm is in the sixth vibhakthi(case). This information is then utilized by the vichcheda module, which ultimately identifies it the word to be rAmasya. The word Acharya is similarly handled with.

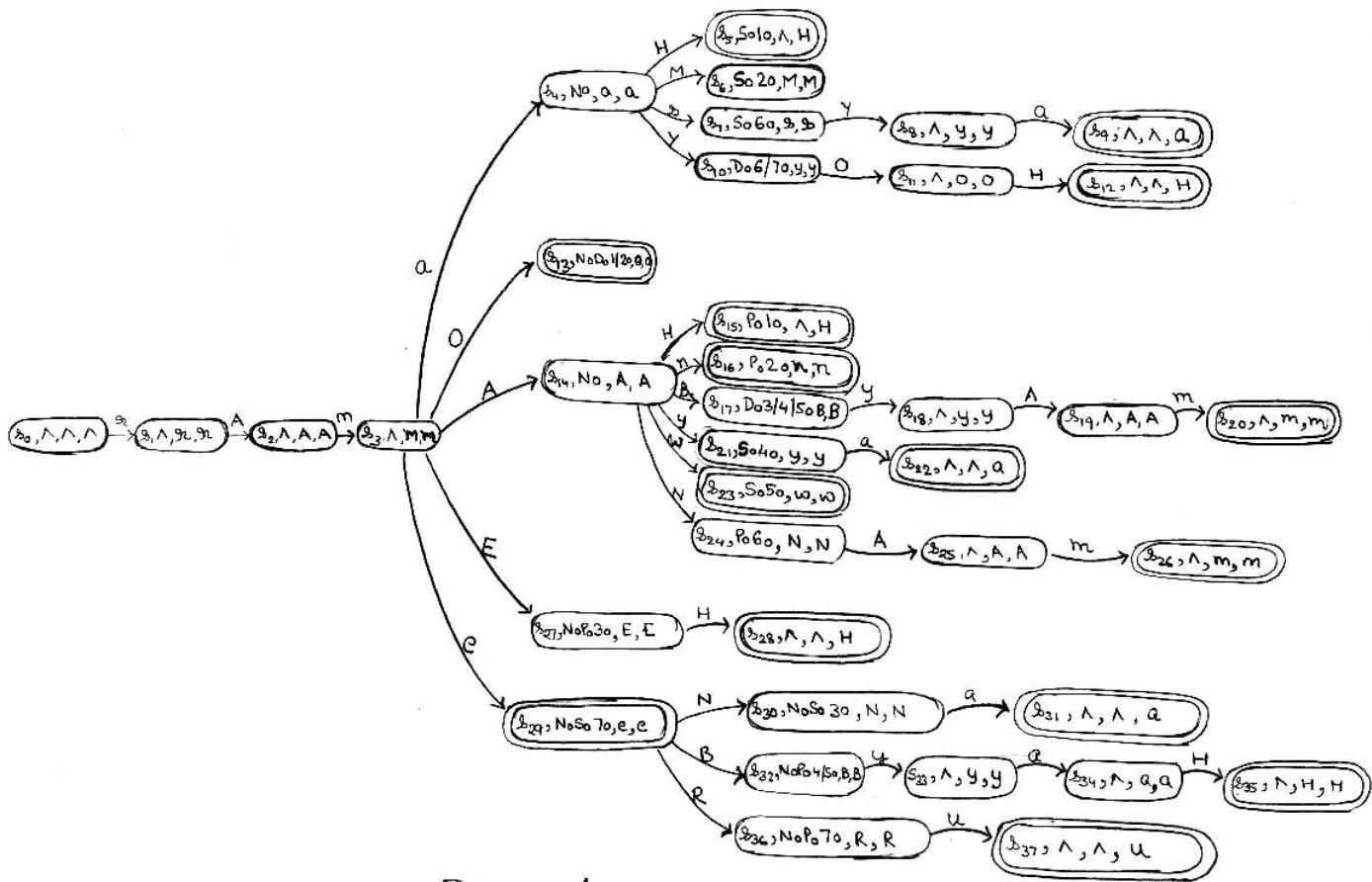


Figure - 4

## Vichcheda Module

This module consists of three parts viz. the Word Producer, the Split Generator and the Sandhi Rule. The module takes the 'partial root' and the immediately following input letter/letters as input. If there are no such input letter/letters then the Word producer module simply transfers the 'partial root' to the translator, else, the Split Generator generates possible split outputs corresponding to the input letter/letters read. It determines the split outputs to be generated according to the Rules of Sandhi found in the Sandhi Rule unit of the Vichcheda module. The Word Producer concatenates the 'partial root' with all the first parts of the pair of outputs generated by the Split Generator module. It sends the concatenated word to the translator. The remaining parts of the pair of outputs are sent back to the vichcheda module. Some rules of the Sandhi Rule module are given below:

$$\begin{aligned} A &\rightarrow a, a \mid a, A \mid A, a \mid A, A \\ I &\rightarrow i, i \mid i, I \mid I, i \mid I, I \end{aligned}$$

U-> u,u | u,U | U,u | U,U  
 e-> a,i | A,i | a,I | A,I | e,a | a,e  
 E-> a,e | A,e | a,E | A,E  
 o-> a,u | A,u | a,U | A,U | o,a | a,o  
 O-> a,o | A,o | a,O | A,O  
 r-> a,q | A,q |  
 y-> i,(vowel) | I,(vowel)  
 v-> u,(vowel) | U,(vowel)  
 ay-> e, (vowel)  
 av-> o, (vowel)  
 Ay-> E, (vowel)  
 Av-> O, (vowel)  
 Where, (vowel) = {a,A,i,I,u,U,e,E,o,O}

Whenever the split generator finds one of the characters on the left hand side of the above rules, it generates all of options on the right hand side of that rule and sends this information to the word producer for each such option. The word producer then concatenates the first part of each option with the output of the parser generator and the next with the rest of the input. If the first input thus obtained is a valid word then that part of the sentence is recognized then and there. The same operation is done on the rest of the string.

## Translator

The translator takes as input the basic words obtained from the vichcheda module and the parse of these words from the parser generator/module and translates these words into their equivalent English words using the knowledge of the parse for these words. The two main jobs of the translator module are to provide a translation for each word of Sanskrit and then to arrange these English phrases in the appropriate sequence.

The 'Structure former' in the present system transforms sentences in any form to the SVO (Subject Verb Object) form. The only restriction of this module is that the Subject and object should appear in the sentence in the same order as that required in the English sentence. This is because both S and O are noun phrases and the distinction between them is difficult.

Translation generator gives the one to one translation of the individual words. It also considers the morphological peculiarities of these individual words. The translation generation makes use of the transducers and a small dictionary giving the English of the peculiarities of each word.

---

## CHAPTER 4 CASE STUDY

The Sanskrit to English translator is a model exemplifying the process of translation between the Sanskrit and the English languages. The system works for a small set of simple words, where the choice of words is done in a manner that is representative of different types of words of Sanskrit. The design of the system has already been discussed in the previous sections. Here, we describe the working of the system as a whole as well as highlight the capabilities and the drawbacks of the system.

The main achievement of the system is the morphological parsing and the sandhi vichcheda being performed. As an example, we take the sentence below:

**puswakaH rAmeNAnIwaH "The book was brought by Rama"**

The sentence is first broken into 'puswakaH', 'rAmeNa' and 'AnIwaH'. The words puswakaH is simply recognized by the transducer but the second word is a compound word of two words and so needs to be broken up. This breaking needs the information stored in the transducer, a grammar specifying the different possible splits that are possible for each concatenating character and a module that concatenate the split given by the grammar with the original words. The concatenating character here is the 'A' between 'rameN' and 'nIwaH'. This 'A' is the input to the split generator which gives the split possibilities as:

**A-> a & a, A-> a & A, A-> A & a, A->A & A.**

Each of these is concatenated with the strings 'rAmeN' and 'nIwaH', giving- 'rAmeNa' and 'anIwaH.'

### **rAmeNa and AnIwaH**

Since, first of the set of words given above does not tally with the knowledge stored in the transducer, it is discarded. The second set of words happens to be correct; hence, the other two options given below are not explored.

'rAmeNa' and 'anIwaH'  
'rAmeNa' and 'anIwaH'

During the split (or, no split, as the case may be), the transducer generates the morphological parse for the option of the split that is correct. This parse in the present

system considers the case, number and the root for each noun (N); the person, number and root for each verb (V); and the meaning for indeclinable (X) words. The parse is used in the generation of the sentence and its structuring. The parse of the above sentence will look something like the following:

N ((THE) BOOK)+1S+NO%N(RAMA)+3S+PRE%X(BROUGHT)%

The '%' above is used as a delimiter to distinguish between the parse of each word. '1S' specifies that the word '(THE) BOOK' has been used in the Nominative (or first) case. The 'NO' indicates that there are no prepositions accompanying the nominative case. The noun 'RAMA' can be similarly interpreted with the change that '3S' means that the noun is a singular in the instrumental case. The notice that the 'THE' has been included in 'BOOK' but not in 'RAMA'. The 'PRE' indicates that the prepositions corresponding to instrumental case i.e. 'BY' or 'FROM' appear before the root i.e. 'RAMA'. We shall see other examples of the use of NO/PRE/POST shortly. Since, 'BROUGHT' is a verb in the past tense, and therefore out of the scope of the system at present, it has been included under indeclinable words. Since brought is a verb in general it has been included to work like one. The above sentence is in the SOV order as far as the structure of the sentence is concerned. The sentence may have been in any order, until the Subject and Object are in the correct order, the ordering of the sentence is reshuffled by the system to give the following output.

### **(THE) BOOK WAS BROUGHT BY/FROM RAMA**

Notice that the ambiguity in the use of prepositions has not been resolved. All options are therefore displayed and it has been left to the user's intelligence to choose the appropriate option. Below we describe yet another example indicating ambiguities of the above sort being handled in a similar fashion.

**saha gacCawi "He goes."**  
**saha liKawi "He writes."**

From the above examples, it is clearly seen that 'go' becomes 'goes' in first person singular but 'write' becomes 'writes' under the same circumstances. Therefore, it is necessary that at some occasions 'es' be added to the verb and at others, 's' be added. Though all such options are displayed, this type of ambiguity is also left to the user to resolve. Thus the above sentences will give the following output:

**HE GO(S/ES).**  
**HE WRITE(S/ES).**

A few more examples of the use of prepositions and affixes the sentence:

**palam vakSAw pawawi** "The fruit falls from the tree."

The parse for the word 'vqkSAw' will be- 'N (THE TREE)+5S+PRE'. The word will therefore lead to translation as- 'THAN/FROM THE TREE'. Though the preposition FROM is more appropriate the word THAN will be appropriate in other situations.

**aham asmi rAmasya miwram**

**I am Rama's friend.**

The word 'rAmasya' is translated to RAMA'S. Here, the ""S" has been added to RAMA because it is in the genitive case.

The shuffling of sentences has been done only for sentences having up to three words one of which is a verb. For the other sentences, which exceed this limit, the sentences have to be given in the right order. An example of a three word sentence is:

**rAvaNaH vAhaneNa gacCawi**

**Ravana goes by the vehicle.**

The Sanskrit sentence is in SOV order with the verb gacCawi(goes) appearing after vAhaneNa(vehicle). The English sentence has to be reframed such that gacCawi appears between rAvaNaH and vAhaneNa. After this restructuring that is done by the system, the output is:

**RAVANA GO(S/ES) BY/FROM (THE) VEHICLE**

---

## CHAPTER 5 CONCLUSIONS

There is a lot that I have learnt from this project. It was my first implementation of a system in the Artificial Intelligence domain. In order to develop the system I had to learn visual PROLOG thoroughly. Also, In order to find the best suitable language for implementation there was a need to thoroughly go through other programming languages and find their pros and cons. I also had the opportunity to interact with leading personalities in India in this field. This added to my experience. While developing the system I devised new techniques and unearthed new problem areas in the translation of a system. Though all of these problems could not be sorted out in the present system, they will be highly useful when this system will be extended and improved.

The aim of undertaking this project was to implement the innovative ideas that had already been presented in my paper on morphological parsing involving sandhi words, earlier and to extend it by including new ideas and enhancing the target domain. This aim was beautifully achieved. Apart from the target new ideas and innovations were also played key roles in the development of the project. Though every effort was made to achieve a complete system a few areas remained unsolved due to their being out of the scope of such a small duration project. While, semantic ambiguities were not handled, pragmatics and discourse considerations were out of the scope of this project. Many ambiguities are left to the user to decide. Some among them are the ambiguities of use of right prepositions and plurals of words. Another aspect that has not been dealt with is the structuring of the sentence. All the above and the few that have not been discussed may form the aim for future research and enhancement of the system.

---

## REFERENCES

1. "Speech and Language Processing- an introduction to Natural Language Processing" by Daniel Jurafsky and James Martin, reprint 2000.
  2. "Natural Language Processing" by Aksar Bharati, Vineet Chaitanya, Rajeev Sangal.
  3. "Introduction to Computer Theory" -second edition by Daniel I.A.Cohen
  4. "A Higher Sanskrit Grammar" by M. R. Kale
  5. "Sandhi Viveka" by A. Varadaraj.
  6. Natural Language Processing using PROLOG" by M. Gerald, M Chris, Addison and Wisley, 1989.
  7. "PRLOG Programming for Artificial Intelligence" by Ivan Bratko, Third Edition, Addison and Wisley 2001.
- 

**FOR A COPY OF THE SOURCE CODE DEVELOPED FOR THIS RESEARCH, PLEASE CONTACT THE AUTHOR.**

**S. Aparna**  
[aparnasubramanian@rediffmail.com](mailto:aparnasubramanian@rediffmail.com)