Chapter 5

DESIGN AND DEVELOPMENT OF THE PROTOTYPE

5.1 Introduction

This chapter discusses the design and development of the prototype machine translation system. The computational model and the data structures selected are clearly described. The major criteria for the selection of these models and data structures were accuracy of translation, speed, space required and ease of future enhancement.

5.2 Selection of Translation model

The major difficulties observed in the development of a translation system for Malayalam to English translation were i) the highly agglutinative nature of Malayalam ii) the wide syntactic structure difference between Malayalam and English. To deal with the huge dictionary size required by agglutinative languages a morpheme based translator was chosen. The dictionary used consists of morphemes and their translations. The survey on the various approaches for machine translation revealed that a direct translation approach can be applicable to only closely related languages. Corpus based approaches also will be inefficient in the case of Malayalam to English translation system because of the wide syntactic difference between the language pairs. Moreover, corpus based approaches require sentence aligned parallel text for each language pair. The transfer based approaches works well for language pairs with wide syntactic difference. Transfer based systems are more reliable, flexible and adaptable in meeting the needs of different levels and depths of syntactic and semantic analysis. So it has been decided to use a syntax transfer
based approach using synchronous tree adjoining grammar (STAG) as the translation approach. The system to be developed is only a prototype and the main objective has been set to check the accuracy of the syntax rules and the transfer rules derived and the accuracy of the final translation.

5.3 Prototype Model

Translation with Synchronous TAGs is a three step process:

1. Derivation tree for source sentence is obtained by parsing the source sentence.
2. Source derivation tree is converted into target derivation tree or trees.
3. Once the target derivation is obtained from step 2, target parse tree is generated by traversing target elementary trees and listing the leaf nodes in the parse tree in order will give the required target string. [100,101,102,103].

The system modules selected are similar to that of SAMPARK system developed by IIIT Hyderabad [14,104]. The syntax structure formalism used is similar to the one used in UCSG based Kannada translator by Murthy. The parsing method is similar to the one in MANTRA system developed by CDAC Mumbai [105]. We have tried to combine modern artificial intelligence techniques with the classical Paninian framework based on Sanskrit grammar [106,107,108,109]. The computational models chosen for the translation, morphological analysis and parsing is a pioneering effort for Malayalam language un attempted by previous researchers. The models chosen are first of their kind in Malayalam language.
5.4 System modules

The major modules in the system are shown in Figure 5.1. There are four main modules:

- Morphological analyser
- Morpheme based parser
- Target sentence generator
- Bilingual dictionary

5.4.1 The morphological analyzer

The analysis module, finds the sequences of morphemes in the input sentence. Since there are more than one way of splitting a word into morphemes it performs a depth first approach to find all possible sequence of morphemes for the given sentence. A Detailed description on the design and development of the module is given in chapter 6. The output of the analyzer module is fed to the parser to create source and target parse trees.
5.4.2 The parser

The parser performs three functions. i) It finds the correct POS tag for each morpheme for word sense disambiguation. ii) It identifies the chunks in the sentence for reordering and creates the source parse tree. iii) It performs the reordering required in order to meet syntactic requirements of the target language and create the target parse tree.
A detailed description of the design and development of the parser module is given in chapter 7. The output of the parser is a set of valid parse trees for a sentence of the source language and their corresponding parse trees in the target language. The target language trees are fed to the target sentence generator module for the final sentence generation in the target language.

5.4.3 Target sentence generator

The function of the target generation module is to generate the target sentence from the target parse tree. It includes the morphological generation module for the target language. This module produces the final English sentence by performing a depth first traversal of the target parse tree created by the parser. It also performs surface form generation of English words using the morpheme sequences in the tree. It also uses additional set of rules for the generation of target sentences in the correct form. It performs two kinds of post processing on the target sentences.

1. It removes the dative case marker ‘to’ from the first noun group (i.e. subject).

   Input: രമത് സുന്‍ (ramakk katayil pOkEnti vannu)

   Rama had to go to the shop

   The depth first traversal of the target parse tree creates the following sentence in English.

   to Rama had to go to shop.

   The initial ‘to’ has to be removed to form the correct sentence as:

   Rama had to go to shop
2. It removes the last ‘and’ in a conjunct noun.

Input: രമാവ് സീതാവ് ചേർക്കാരുന്ന് (ramayum seethayum koottukaaraaN)

Rama and Seetha are friends

The depth first traversal of the target parse tree creates the following English sentence

Rama and Seetha and are friends

The last and in the noun group is eliminated forming the correct sentence as

Rama and Seetha are friends

5.4.4 Bilingual dictionary

The bilingual dictionary is the most essential part of the whole machine translation system. The first rule based translation system Systran uses for every source language two types of related dictionary: stem dictionary and expression dictionary. The stem dictionary contains the basic form of single words and expression dictionary contains idiom replace, collocation, conditional expression, etc. For the present system only one dictionary had been kept as it was designed for a prototype machine translator. The major concerns were to develop and test the functioning of the analyzer and the parser module which were unattempted so far for Malayalam language. The structure of the bilingual dictionary of our prototype system is shown in Table 5.1. Excerpts from the bilingual dictionary are given in Appendix 1. The bilingual dictionary contains the following information:
i) The dictionary includes most of the commonly occurring verbs, nouns, pronouns, adjectives, inflectional and derivational suffixes, clause suffixes etc.

ii) Each entry in the file has four fields: the root word (morpheme), the morpheme tag, category (human/nonhuman) and its translation. A root word can have more than one tag.

iii) The verbs in past tense have their root words stored along with them.

Table 5.1 Bilingual dictionary

<table>
<thead>
<tr>
<th>Source word</th>
<th>Morpheme tag</th>
<th>Target word</th>
<th>Root word</th>
<th>Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>꾌</td>
<td>Noun (N)</td>
<td>cat</td>
<td></td>
<td>Non human</td>
</tr>
<tr>
<td>꾈</td>
<td>Case suffix (NA)</td>
<td>‘s</td>
<td></td>
<td></td>
</tr>
<tr>
<td>꾌</td>
<td>Verb (V)</td>
<td>went</td>
<td>go</td>
<td></td>
</tr>
</tbody>
</table>

It has been found that a trie model for dictionaries helps in fast retrieval of data. A prototype dictionary had been implemented using trie model. The trie had been implemented as a two dimensional array. Though the system gave 95% saving in retrieval time the space requirement was 93% higher than that of a linear storage of data in a text file. Since the main memory space is scarce more advanced data structures have to be used to implement the trie model. Taking these factors into consideration the dictionary was implemented as a sequential file containing the morphemes, their lexical categories and their translations.
5.4 Implementation

The modules were separately developed in Python language and tested separately for accuracy. After sufficient amount of testing all the modules were combined to build the prototype. The system can work in Linux or Windows operating system. The input sentences were stored in a text file. The set of grammar rules in the context free notation and bilingual dictionary are also stored as text files. The structure transfer rules were implemented in the source code of the parser.

5.5 Conclusion

This chapter discussed the design and development of the prototype machine translation system. The major modules used in the system were explained in the chapter. The post processing steps done by the target sentence generator module and the bilingual dictionary structure were also given. The next chapter gives the computational model chosen for the morphological analyzer, the working of the algorithm for the analyzer and the data structure used for the splitting rule table.